TEL AVIV UNIVERSITY

The Iby and Aladar Fleischman Faculty of Engineering
The Zandman-Slaner School of Graduate Studies

Estimation of Tomato Leaves Orientation for Early Diseases Detection Using Deep Neural Network Model

A thesis submitted toward a degree of
Master of Science in Mechanical Engineering
by:

Adi Cohen

October 2022

TEL AVIV UNIVERSITY

The Iby and Aladar Fleischman Faculty of Engineering
The Zandman-Slaner School of Graduate Studies

Estimation of Tomato Leaves Orientation for Early Diseases Detection Using Deep Neural Network Model

A thesis submitted toward a degree of
Master of Science in Mechanical Engineering
by:

Adi Cohen

This research was carried out in The
School of Mechanical Engineering
This work was carried out under the supervision of
Prof. Avital Bechar, Dr. Avishai Sintov

October 2022

Acknowledgment

First, I would like to thank my supervisors, Prof. Avital Bechar and Dr. Avishai Sintov, for their professional guidance, insights and support they have given me since the very beginning of my research.

I gratefully thank the Volcani Institute – ARL and Tel Aviv University for their financial support during the years 2021-2022.

Also, I would like to thank the members in ARL (Agriculture Robotic Lab) at the Institute of Agricultural Engineering for their help and advice. A special thanks to Yarden Akaby for a great contribution in the research and the experiment apparatus, and Asi Lazar for professional guidance.

Finally, I would like to thank my family and friends, for their encouragement, and infinite support.

Abstract

Early detection of diseases in tomato is essential to prevent infection of fruit and other plant parts, to minimize spreading of the disease in the plot and to reduce crop losses and maintain quality. A robotic system equipped with an electric resistance sensor is developed to reach, grasp, and measure the conductivity of a tomato leaf. The electric resistance is proven to be a significant indicator regarding the plant's condition in early stages of the disease. To grasp a leaf and perform the measurement with a robotic arm, a method for automatically identifying the leaf orientation was developed using data from images of leaves as input to a Deep Neural-Network (DNN) model. An experiment was conducted using an UR5 robotic arm to collect and label leaf images at different orientations and distances. Each image was processed to extract an array containing a list of boundary points of the leaf as well as information of the manipulator position and orientation, and this data is then used as input for the DNN model. The measured information of the robotic position, and the primary known orientation of the first image taken for each leaf, are used to label the "ground truth" orientation with respect to the camera. These labels are used in the process of training the DNN. The DNN then receives the input and produces a spatial vector normal to the leaf's surface. To determine the leaf orientation, several analyses were conducted using one, two and three images of the same leaf in different poses and the differences in the manipulator orientation to take them. The main results show an average accuracy of 11.5 degrees between predicted and real orientation of the leaf's surface which, in most cases, it is sufficient to grasp the leaf and perform the measurement. In addition, the DNN model results were analyzed to optimize the robotic motion to best predict the leaf's orientation and improve the accuracy. Those results are satisfying for the robotic end effector to perform the measurement of the leaf's resistance.

Table of content

List of Figures

List of Tables

1. Introduction

1.1 Problem and Motivation

Tomato plants are grown in protected facilities such as greenhouses and screen-houses. These facilities are favorable for infections by various types of viruses. One of them is the "Tomato Brown Rugose Fruit Virus" (ToBRFV)- in the genus ’Tobamovirus’ (Luria et al., 2017) . This virus is highly aggressive and can cause severe damage to tomato plants. Leaf symptoms of the ToBRFV include wrinkling and bubbling with an accompanying mosaic pattern. The tomato fruit has a browning calyx and is undersized with a rough surface (Fig. 1). ToBRFV causes the remaining fruit to be blotchy, pale and may have brown, necrotic spots. Plants infected early will produce a poorly formed fruit, while plants infected later may not express fruit symptoms until the fruit turns red. (*Tomato Brown Rugose Fruit Virus (ToBRFV): A New Concern for Tomato and Pepper Producers - Vegetables*, n.d.). The Tobamovirus can cause great economic loses. The impact of these plant diseases on yield can be devastating and lead to great losses of the crop or damage its quality.

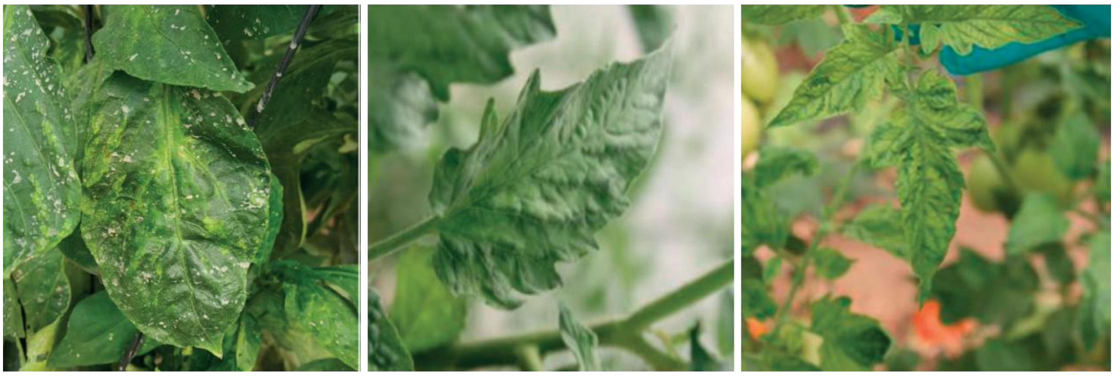

Fig. 1. : Symptoms caused by ToBRFV include bubbling and mosaic on leaves of susceptible pepper, and fern leaf and mosaic of tomato (*Tomato Brown Rugose Fruit Virus (ToBRFV): A New Concern for Tomato and Pepper Producers - Vegetables*, n.d.).

Tobamovirus is transmitted between plants by contact and is being spread within the plot. Such contact is caused by a mechanical transfer of the virus on the hands, cloths and tools of the workers in the field, unintentionally. Furthermore, the Tobamovirus can be transmitted even to seeds (Schor et al., 2016), so that 10-16 days after planting them, when agrotechnical procedures are made to design the plants, a second spread of the disease may occur. The Tobamovirus contaminating has facilitated the spread of the viruses to various parts of the world (Fig. 2). (Kenyon et al., 2014; Ling et al., 2019; Zhang et al., 2022)

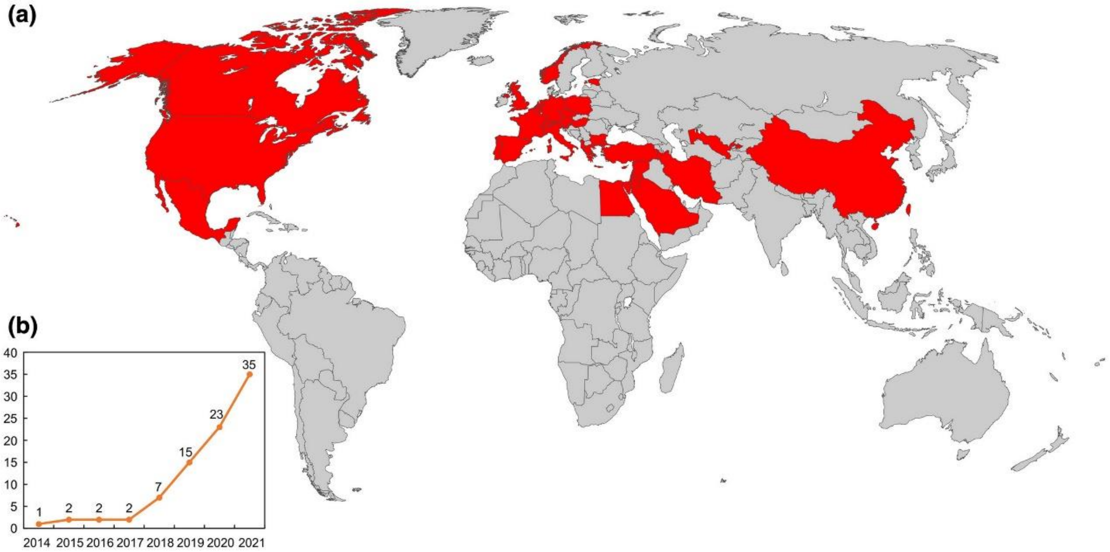

Fig. 2. Distribution of ToBRFV. (a) The geographic map of ToBRFV. (b) The accumulated number of reported countries starting from the first outbreak in 2014. (Tomato Brown Rugose Fruit Virus (TOBRFV)[World Distribution]| EPPO Global Database, n.d.)

ToBRFV tendency to spread easily among other plants, explains the importance of early detection for ensuring successful prevention of the disease and applying counter measures. Currently, no commercial ToBRFV resistant cultivars are available (Zhang et al., 2022). Therefore, periodic, and repetitive monitoring is essential to locate symptomatic plants and remove them to prevent infection of others. Today, expert inspectors manually conduct the monitoring process to prevent diseases and the detection of those diseases is completely dependent on the observation experience of the grower. In this process, the sampling spatial resolution and temporal frequency are low (Schor et al., 2016). This method is slow, have low efficiency, high cost and is time consuming.

With the continuous development of technology, new and advanced methods for diseases detection are evolving. Using those methods and improving disease detection procedures is a critical factor in increasing greenhouse productivity, quality, and sustainability (Liu & Wang, 2020; Schor et al., 2016). Automation and optimization methods of manual procedures have been proven to improve the productivity of agricultural machinery by increasing efficiency, reliability, and precision, and reducing the need for human intervention. (Bechar & Vigneault, 2016). There for, automatic and advanced technologies are expected to improve disease detection procedures in early stages of growing the plants. This improvement can help coping with these difficulties, decrease the sources of the disease and, therefore, improve yield production and lower the use of pesticides. (Durmu et al., n.d.; Fountas et al., 2020; Schor et al., 2017).

In this context, integrating robotic automatic systems into various agricultural fields tasks calls for a reductions in labor and workload, as they can enable collecting and analyzing data in an efficient, cost-effective matter. Especially in tasks of a repetitive plant monitoring for purposes of disease detection. Robots can be equipped with different sensors such as optical sensors, current sensors, temperature, or humidity sensors and so on. Those sensors, combined with an automatic system, can collect a layered information about the environment, analyze this information and make "real time" decisions based on the analysis and the results. (Ampatzidis et al., 2017; Bechar & Vigneault, 2016; Fountas et al., 2020; Oliveira et al., 2021)

Developing advanced sensors for disease detection and integrating them into an automatic robotic system, is a promising approach. A novel new method for ToBRFV early detection was developed (Reddy et al., 2022). This method relies on the changes in the electrical properties of the tomato leaves caused by a development of the disease. The electric resistance and currents were proven to be a significant indicator of early stages of the disease when the visual symptoms are not yet to be recognized and can be measured by an electric resistance sensor.

Inspired by the many advantages of automatic robotic application for agricultural assignments, a robotic system equipped with this electric resistance sensor is developed to reach, grasp, and measure the conductivity of a tomato leaf using a sensor that applies voltage and measures the current of the leaf. Despite the vast research and use in the field of robotic agriculture, there is limited information in the literature about monitoring plants like tomatoes in protected structures for disease detection (Fountas et al., 2020) and are mostly about data acquisition and analysis. Furthermore, the use of nanotechnology such as using a current sensor in agriculture is limited. Robotic sensing for the purpose of early detection of diseases in plants is innovative. The robotic system in this method of sensing is necessary since the measurements require an accurate movement for grasping and attaching the sensor to the leaf surface. A significant challenge of planning the robotic manipulation to the leaf's surface is detecting the accurate orientation of the 'target' tomato leaf.

1.2 Objectives

The main objective of this study is to develop a method to estimate the orientation of a given leaf (on the plant) in relation to the robotic arm coordinate system, based on information from images of the leaf, taken with an RGB camera placed on the robotic end effector. The objective can be separated to two specific objectives:

- **Develop a model to estimate the leaf surface orientation given images of the leaf**

  In this study, the process of extracting relevant features from this raw data is determined, along with the development of a suitable model for estimating the leaf orientation using the processed data. Models for estimating a tomato leaf orientation for purposes of robot manipulation are highly limited in the literature so this research provides a first direct solution for this task

- **Determine a technique to use the robotic arm and the trained model in "real time" for the optimal estimation of the orientation.**

  Looking ahead for the implementation of the model in a functional automatic robotic system, it is highly important to determine the optimal methodology for using the model in a way that will maximize the accuracy in "real time". Therefore, the optimal route of the robotic arm is planned for achieving the highest probability to improve the estimation accuracy.

1.3 Approach

In this research, detecting the accurate orientation of the 'target' tomato leaf is required. A method to complete this task is developed, using two main components: a robotic arm and an RGB camera attached to it. The main challenge in this research is converting 2-dimmentioal (2D) information from images to a spatial representation of the leaf orientation. An algorithm is developed to make an accurate estimation. The main approach for this algorithm includes extracting information from images taken by the camera and combining it with information of the robotic arm pose when taking those images. This data will be the basis for training a deep learning model.

2. Literature Review

2.1 Precision Agriculture Technologies

Precision agriculture (PA) is a concept based on sensing or observing and responding with management actions to variability in crops. As described in Q. Zhang book, machine automation is a key component in PA. Automation cannot only improve productivity but also solve problems related to limited labor availability and high labor costs due to the aging farm population. Robotic farming, combined with an intelligent automatic control system, will continue to be the focus of the agricultural industry for its promising advantages. Precision agriculture has two phases: collecting the data and analyzing it for practical purposes. Those phases require special equipment and software to collect and analyze all the information. Technologies involving GPS and GIS are used for mapping of irrigation systems and fields (Palaniswami et al., 2011), as well as detection of problematic areas and farm management. Mobile devices such as UAV (unmanned aerial vehicles) or drones can be used to monitor the fields conditions without physically arriving to the area (Puri et al., 2017; Tsouros et al., 2019). In addition, sensors can be attached to them, such as multispectral, thermal or visible cameras, to record needed information (Daponte et al., 2019).

Different types of sensors have been used for precision agriculture to gather data on soil water availability, soil compaction, soil fertility, leaf temperature, leaf area index, plant water status, local climate data, insect-disease-weed infestation, and more. Optical sensors, for example, can be positioned on automobiles, satellites, drones or robots to record required information (Viana et al., 2019). Temperature sensors are important for purposes of ambient condition monitoring(Palazzi et al., 2019). Electromagnetic and dielectric sensors aid in the collection, processing, and mapping of the chemical data of the soil such as the nutrient of the soil levels, pH, and moisture levels in the soil (Yin et al., 2021). They are usually mounted on specially designated areas.

The following sections will address the uses of both robotics in agriculture, as well as artificial intelligence (AI) applications used in this field.

2.2 Robotics in Agriculture

The development of new technologies allows using autonomous robotic systems for a wide range of agricultural tasks and procedures. The use of autonomous robots reduces human resources required for a given surface, defines tasks and can improve the accuracy of plants monitoring compared to expert inspectors. Agriculture tasks that can be improved by an autonomous robots include planting, accurate and automatic sowing, spraying, and harvesting. (Bechar & Vigneault, 2016, 2017; Fountas et al., 2020; Oliveira et al., 2021). Oberti et al., (2016) has demonstrated, for example, selective spraying of grapevines using an agricultural robot, by using a precise spraying end-effector mounted on the robot integrated with a disease-sensing system based on multispectral imaging. Barth et al., (2016) have developed a sensing and servo control framework for harvesting, that allows flexible implementation of sensing and motion control using robotic systems. Luo et al., (2016) combined computer vision methods and a robotic system to detect spatial coordinates of the cutting points on a peduncle of grape clusters by an end- effector.

Robotic systems are also used for monitoring and "blind" driving in field. Those systems face many challenges by their nature of working in dynamic, unknown environments. Facing this challenge, Bac et al., (2016) developed an algorithm to evaluate the effect of different plant and robot parameters on the detection of collision free path for a robotic system. Those parameters are stem spacing and fruit location and the end-effector dimensions as well as the robot position.

Robotic applications are developing in the field of plants disease detection as well, as they integrate the use of sensors to detect and classify different diseases. So far, for disease detection purposes, robots have included a vision-based systems. (Fountas et al., 2020). Schor et al., (2017) for example, developed a robotic detection system for two major threats of bell pepper plants, using optical sensors mounted on the robotic arm. Disease detection methods will be elaborated more on the following sections.

2.3Artificial Intelligence in Agriculture

Artificial intelligence (AI) is intelligence demonstrated by machines. The applications of AI methods are used in a variety of areas. In the agriculture field, farming solutions which are AI powered enables a farmer to do more with less, enhancing the quality of the crops. AI application have been used for various tasks such as disease detection, field management, automatic irrigation techniques, and crop monitoring. The methods used under the term "AI" include machine learning (ML) (K-means, support vector machines (SVM), and artificial neural networks (ANN). A method that is gaining momentum is deep learning (DL).

For example, Bhaduri et al., (2009) demonstrate an application of machine learning in agriculture as they developed an endmember extraction algorithm based on SVM to the problem of estimation of vegetation endmembers from a hyperspectral image. In addition, Tellaeche et al., (2007) showed that k-means classifiers can support a hybrid decision making system for detection and differential spraying of a weed growing in cereal crops.

Deep learning is the state-of-the-art machine learning method, that utilizes artificial neural networks (ANNs) with hidden layers. This allows larger learning capabilities and thus higher performance and precision. (Dharmaraj & Vijayanand, 2018; Kamilaris & Prenafeta-Boldú, 2018). Most methods for the said tasks, are using convolutional neural networks (CNN, type of DL models) that ate more suitable for image processing and classifications. Deep learning applications in agriculture mostly deal with image classification and identification of areas of interest, including detection of obstacles or fruit counting (Chen et al., 2017). For example, Nguyen et al., (2020) developed an autonomous system to differentiate crop areas from non-crop areas. This is done by a deep neural network trained on imagery data streams from low-Earth orbiting satellites. Another use of deep learning is presented in the work of Steen et al., (2016), for obstacle detection in agricultural fields. This is done by training a model on RGB images of obstacles in different backgrounds.

Other areas of deep learning agriculture address future predictions of the quality of the crop and crop yield estimation, like the work of Elavarasan & Durairaj Vincent, (2020) that used a deep recurrent network to forecast the crop yield. This is done by training the model over a dataset that incorporates specific climate, soil, and groundwater properties along with the volume of fertilizers consumed by the crops. Weather prediction has great value in the agriculture area and is one of the most experimentally difficult problem in the world. A deep learning method was developed to face this task as well (Mohan & Patil, 2018), using dimensionality reducing strategy of the weather-related large datasets, to improve the accuracy of whether predictions. A different method is used for land cover classification task in agriculture. This task is addressed in the research of Kussul et al., (2017) that targets land cover and crop type classification from multitemporal satellite imagery and a deep neural network.

The number of deep learning applications in agriculture is very large (Kamilaris & Prenafeta-Boldú, 2018; Zhu et al., 2018), as only a small part of the examples is detailed in this review. Sections 2.4, 2.6 addressing plants disease detection and leaf pose estimation, demonstrate a detailed use of deep learning methods in order to address these subjects.

2.4 Plants Disease Detection - Traditional Methods

To date, the techniques used for detecting plant disease are divided into direct and indirect methods (Fuentes et al., 2016; Sankaran et al., 2010). Direct methods use chemical analysis of the infected area such as serological methods or molecular techniques. The sensitivity of the molecular techniques refers to the minimum amount of micro-organism that can be detected in the sample. Those are presently the most accurate methods for plant disease detection, yet their limitations are that they are destructive tests, time-consuming, labor-intensive. Usually, those methods are not performed in real-time and require specific instrumentation.

Indirect methods are more common due to their non-destructive approach. They use extrinsic techniques such as spectroscopic and image techniques. The non-destructive approach approaches use information from fluorescence, multi-spectral (Oberti et al., 2016), hyper-spectral (Mishra et al., 2020) or spectroscopic images, to detect and classify the different diseases.

Using fluorescence imaging is done by analyzing the changes in blue-green fluorescence and chlorophyll fluorescence that can indicate the plant's physiological condition. This method can distinguish between infected and non-infected plants. However, it is not effective for estimating the state of mildly infected plants. Multi-spectral and hyper-spectral simulations are used to analyze the different pixel values changed in different wavelengths of the spectrum. The challenges of these approaches are selecting the specific spectral band and classification algorithm for a specific disease.(Sankaran et al., 2010)

Spectroscopic images mostly use RGB images by extracting valid features from the images and developing detection algorithms. For example, research by Schor et al., 2017 have showed the importance of combining an RGB sensor along other sensors such as multispectral sensors to detect diseases in pepper plants. This method integrating a PCA detection algorithm that classify every pixel as healthy or diseased, and marking the diseased areas. This method achieved accuracy of approximately 85.6% and 95.2% detecting the 2 types of pepper plant diseases. It is important to emphasis, that this method is used to determine whether the plant is healthy or diseased, without detection of the disease stage.

2.5 CNN Models in Agricultural

Training convolutional neuron network (CNN) models has become an efficient tool for the task of disease detection and classification. Using this method mostly involves collecting a large database of images to be examined and labeled for the training process. Deepak et al., 2019 research for example, demonstrated a CNN model that was trained to improve the image classification of three classes of tomato plants conditions: Leaf curl diseases and Septorial Spot diseases as well as healthy leaves. The dataset used for the training of the model is Plant Village Dataset from the Plant AI competition, as well as a customized dataset, and the model yielded an accuracy of 92% for the classification.

Another method (Durmu et al., n.d.) modified existing CNN models (AlexNet and SqueezeNet) that were proven to perform image classification well, to the use of tomato leaves disease classification. Like the work of Deepak et al., 2019 , PlantVillage dataset is also used in this method. Yet AlexNet and SqueezeNet models have achieved better accuracy of 95% and 94%, correspondingly.

Using CNN models over images taken in field environment is a complicated task, since they mostly have a complex background or may be taken under non-optimal conditions of light and camera angle. Liu & Wang, 2020 work have created a customized dataset of tomato diseases and pests images under the real natural environment. In this work, CNN trained model was also used – as an improved "Yolo V3" model that was tested for classification of 12 different types of tomato disease and pests. The accuracy achieved for this method is 92%. In the work of Sladojevic et al., 2016 a model developed was able to recognize 13 different types of plant diseases out of healthy leaves, with the ability to distinguish plant leaves from their surroundings. In this work, a search online for different images of diseases fruit leaves (apples, pears, peaches and grapevines) was made to create the dataset, as well as augmentation and integration of different backgrounds of the images. The model trained is an existing CNN model- "CaffeNet", that was modified to support the classification task. This process yielded a 96.3% accuracy.

As shown with those few examples, CNN models perform image classification and image processing tasks with ease as they cope with high dimensional images. In addition, those methods have been proven to achieve better accuracy for a large variety of images. Yet, the use of RGB images as input for the network is incomplete. RGB images as disease indicator major drawback is that it relies on visual symptoms of the disease that are recognizable only after the disease has developed for a few days.

2.6 Leaf Pose Estimation

A significant challenge of executing an accurate movement for grasping and attaching the sensor to the leaf surface is detecting the accurate orientation of the ’target’ tomato leaf. The literature regarding object pose estimation is wide and mostly address this problem with methods using CNN trained to estimate the pose of specific objects. Estimating the pose of leaves in the 3-dimensional (3D) space is a challenging problem due to their different shapes and appearance in images that can be affected by lightning conditions, clutter in the scene and occlusion between objects. The literature regarding the specific task of leaf orientation detection is very limited.

2.6.1 Object pose estimation- methods and techniques

Classical techniques addressing the process of object pose estimation mostly use feature extraction from a 2-dimensional (2D) RGB image and match them with a known 3D model to align those models with the 2D image. The pose estimation for these objects is computed by Perspective-n-Point (PnP) algorithm (REF). Those methods require a rich texture of the objects to be able to match the correct features to their relevant location in the 3D model. Some of these methods integrate "Random Sample Consensus" (RANSAC) algorithm to avoid those outliers in the process. Other methods integrate depth information to improve the pose estimation. Depth information addition is also used to recognize the object pose even without an available 3d model of the object (Wang et al., 2019). Better performing techniques use CNN trained for the task of fitting pre-scanned 3D object models to the resulting segmentation to get the object pose (Park et al., 2019; Xiang et al., 2017; Zappel et al., 2021). Hu et al. (2018), for example, demonstrated a segmentation algorithm that is matching visible parts of the objects to a local pose prediction, in a form of 2-dimensional (2D) key points. The local pose estimation is done by a CNN model. Every image is divided to small patches, so that each patch is matched with an estimation of a projection of a given 3-dimensional (3D) model of the object in the image.

Zeng et al. (2017) work has presented a self-supervised model for 6D pose estimation in a cluttered environment for purposes of grasping objects with a robotic arm. This is done by segmentation of RGB-D point clouds captured from multiple views into different objects, using deep convolutional neural network, and then aligns pre-scanned 3D models of the identified objects to the segmented point clouds to estimate the 6D pose of each object. Those methods enable dealing with the challenge of occlusion yet require an existing 3D model of the object.

As described by the various examples of pose estimation, most methods require a large dataset of images, or additional data such as depth, or 3D cad model of the target objects.

2.6.2 Leaf orientation estimation- methods and techniques

Specific research on leaf orientation estimation is limited. Most of the related work use image processing or deep learning methods for tasks different then finding the orientation of the leaf. This includes, for instance, detecting the leaf closest to the camera, or the best location on the plant for de-leafing and harvesting by a robotic system. Barth et al., (2019) estimated the angle between plant parts for purposes of harvesting. This was done by training a CNN trained model for segmentation of images distinguishing the stem from the fruit, and then calculating the stem angle by using image processing methods such as edge detection and hough transform.

Other methods were used to detect a location of the leaf. Ahlin et al., (2016) proposed a method for autonomous leaf picking using deep learning and image processing. The detection and segmentation of the leaves was solved by training a CNN model on a dataset of cropped images of plants labeled with classes of leaves or background.

Beadle et al., (2020) demonstrated another method for position estimation of leaves with expected error of 20 mm, this method is computer-vision based for estimating the three-dimensional location of all leaves of a subject plant with the use of a single digital camera autonomously positioned by a three-axis linear robot. A custom trained neural network was used to classify leaves captured in multiple images taken of a subject plant, and calculations were applied using several images to predict the leaf's depth, and from this, the three-dimensional position.

Jeon & Rhee, (2017) have proposed a method for leaf types classification using a CNN model "GoogleNet". The model was trained on " Flavia dataset" including 9 different types of leaves and yielded an accuracy of more than 98%. Segmentation challenge of multiple leaves in one image has also been addressed in a number of other researches (Scharr et al., n.d.). One method (Alenyà et al., n.d.) offered segmentation that relies in the infrared intensity image, and the depth information that is received by the plant images. Those images are segmented to surfaces as the leaf contour are fitted to the extracted segments. Another method (Khan & Debnath, 2019) propose techniques for processing an image to obtain contours of every individual objects. Then, it selects the best appropriate connected contours that represent region of every leaf appearing in an image. This model achieved an overall 90.46% segmentation rate.

Neither of the methods provide an actual solution for the leaf orientation estimation task, but the different techniques emphasize the efficiency of image processing and the importance of collecting a customized database for a specific task.

3. Materials and Methods

3.1 Overview

The objective is determining the leaf orientation in order to be able to grasp it using the robotic manipulator. Grasping and measuring the leaf will be accomplished using a UR5 robotic arm, a gripper (developed in parallel research, Gorali at el.,2022) and an "RGB" camera attached to the robotic manipulator. The gripper used for measuring of the leaf's electric properties, allows tolerance of 20° in the orientation of the leaf to make a satisfying measurement.

The approach proposed for the task of leaf orientation estimation is using deep neural networks (DL) to train over a labeled dataset of leaf images in different poses to predict the spatial vector, normal to the leaf's surface in relation to the robotic manipulator. Then, plan the route of the robotic arm to grasp and measure the electric properties of the leaf. This method simulates one, isolated, target leaf as if it is a part of a whole tomato plant, as methods for leaf segmentation have been developed (Khan & Debnath, 2019) and will be integrated in the complete system. In addition, the leaf's curvature surface is approximated to be a flat plane. When the orientation of the leaf is defined to be the normal vector of this plane, as it is represented in the camera (attached to the robotic manipulator) coordinate system. Given this normal vector, manipulating the robot to this orientation is possible.

In order to train a DL model to estimate the leaf orientation, a relevant database is collected and labeled. The input for the network will be an array containing a 2D matrix of leaf boundary points extracted from leaf images. The required labels (the "ground truth") for each input are a 3 X 1 vectors representing the normal vector perpendicular to the leaf surface, represented in the robotic end effector coordinate system. As the data is collected using the robotic arm with a RGB camera mounted on it, the pose of the robotic arm when taking every image will be saved as additional data. In addition, all images will be saved with its relevant parameters such as the distance of the camera from the leaf, and the leaf orientation.

In this research data acquisition is required since a relevant dataset does not exist. The process includes collecting a large database of images of tomato leaves in different orientations. The vector perpendicular to the leaf surface is then calculated in relation to the coordinate system of the camera. In order to complete the data acquisition process, an experiment apparatus is designed to automatically collect the images. In addition, an automatic labeling method is developed.

3.1.1 Research General Steps

To accomplish the purposes of this research, the method involves these main steps:

1. Experiment design:
    a. Creating an experiment to collect a database of leaves in different orientation, using UR5 robotic arm, and RGB camera.
    b. Defining the experiment variables and coordinate system
2. Data acquisition:
    a. Colleting of a database of leaf images
    b. Developing a method for labeling each image with the equivalent orientation
3. Data processing:
    a. Image processing and features extraction
    b. Arranging different groups of databases for model training: (datasets that contains different number of images of the same leaf as input, and datasets that contain an additional information of the robotic movement)
4. Model training:
    a. Developing and modifying a DL model for estimation of leaf orientation
    b. Training the model over different datasets and hyperparameters
5. Model analysis:
    a. Detecting optimal set of databases and hyperparameters
    b. Estimating the optimal route of the robot to make in "real time"
6. Evaluation - Conducting an experiment for validating the model and concepts

3.1.2 "Human pose" method as inspiration

In this research, RGB images (without depth information) are the main form of information used. The inspiration for this research approach is a method used for estimation of a human pose (Martinez et al., 2017). The method uses the pixel locations of 16 joints of the human body as they appear in the image and predicts the spatial pose of the human in the world. This method is similarly facing the challenge of extracting spatial information from 2D RGB images, by using a simple, deep, multilayered neural network with batch normalization, dropout, and residual connections. This approach is effective and relatively easy to compute, as well as it does not rely on the image properties such as lightning and occlusions but only accepts the relevant 2d positions and extract the desired 3D pose. Fig. 3 displays the architecture of the network (Martinez et al., 2017).

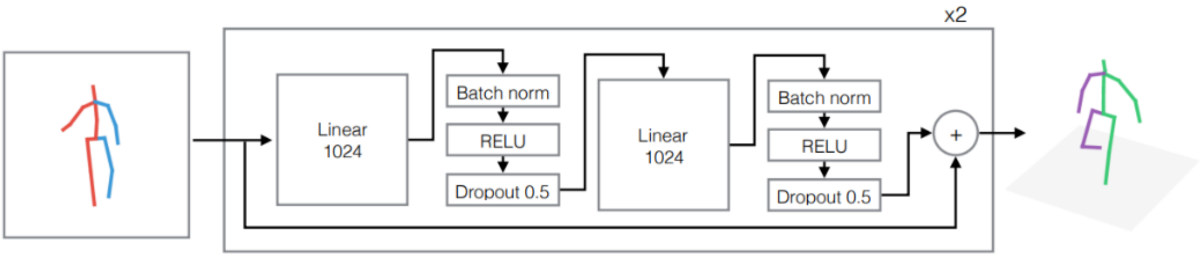

Fig. 3. 3D human pose estimation- architecture of the network (Martinez et al., 2017)

3.2 Experiment Design

A large database of 15,840 labeled images of tomato leaves in different orientations is collected as part of the research, by an experiment apparatus built for this task. The experiment apparatus includes a UR5 robotic arm, an RGB camera mounted on the robot end effector, a turntable placed in front of the robotic arm and a pole holding a mechanism for placing the tomato leaf, that is connected to the turntable center. Fig. 4 illustrates the simulation of the experiment apparatus combined and Fig. 5 shows the actual experiment apparatus as built.

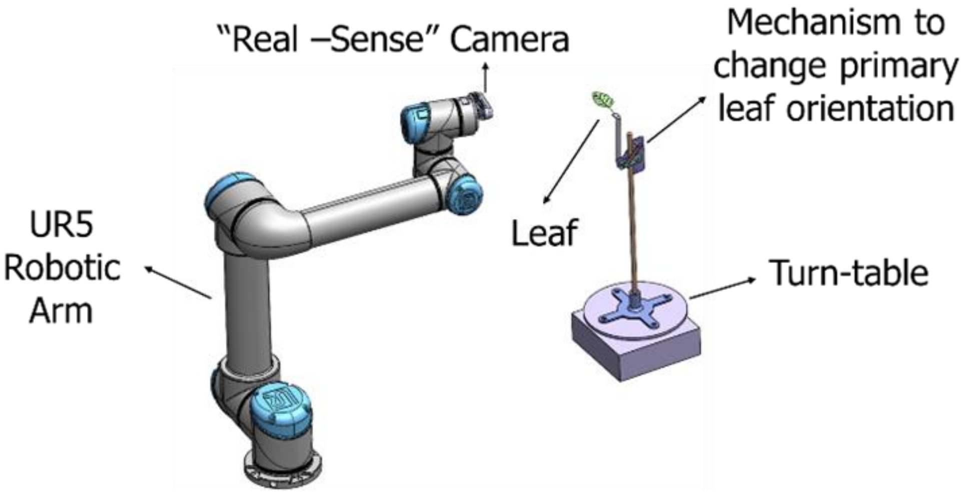

Fig. 4. Experiment Apparatus – Simulated

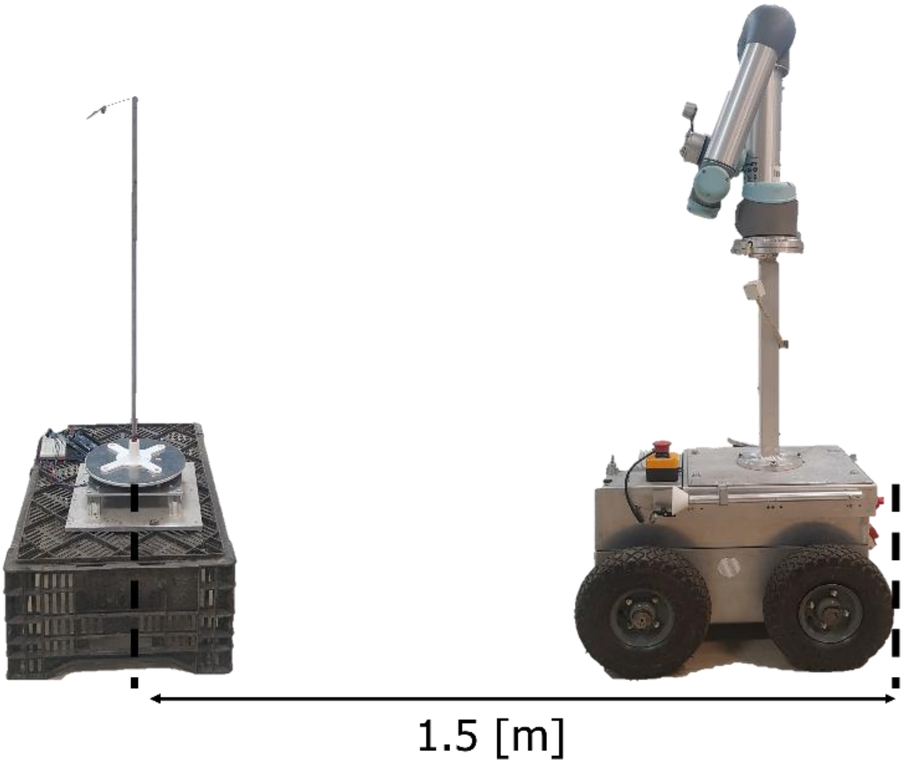

Fig. 5. Experiment Apparatus in practice

3.2.1 UR5 robotic arm

The Universal Robots UR5 is an industrial robotic arm designed to simulate repetitive manual tasks. With 6 degrees of freedom (DOF) and a carrying capacity of 5 kg and a radius of 850 mm. In this research, the robotic arm is used for both data acquisition process as well as real time image collecting to validate the models results. Fig. 6 illustrates the UR5 robotic arm.

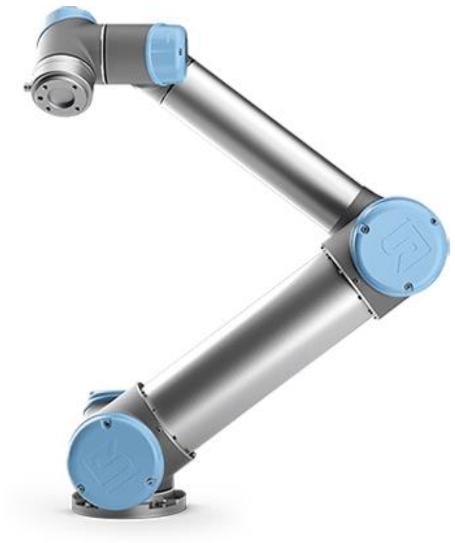

Fig. 6. UR5 Robotic Arm

Controlling the UR5 robotic arm, in this research is done by using Python socket module. Sockets and the socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The connection is done first by defining "host" – in this case is the IP address of the UR5 robotic arm, and the "port" - is the open port on the UR which listens for the commands sent from the computer. Socket module enables the connection between the host and the port, to control the robotic motion.

3.2.2 Camera

In this research, the camera used for the data acquisition process is an Intel “RealSense” d435i camera seen in Fig. 7. This camera is used as an RGB camera.

Fig. 7. Intel “RealSense” d435i camera

3.2.3 Turntable

The turntable (developed in parallel research, Akaby at el. 2022) is a mechanism combined with a rotating plate connected to a step-motor and Arduino controller. This mechanism enables the plate to rotate in a desired angle every step. In this research, the turntable was programmed to rotate an angle of 30° each step, after 12 steps, the turntable completes a full circle. Fig. 8 shows the turntable.

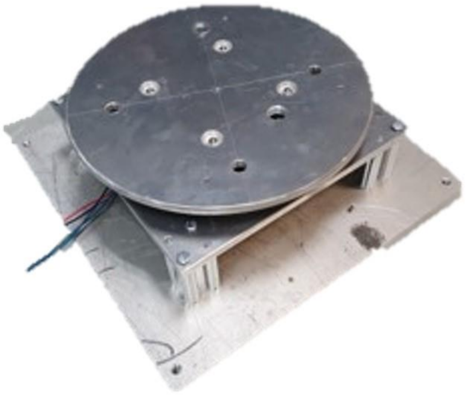

Fig. 8. Turntable

3.2.4 Leaf holder mechanism

The leaf holder mechanism solution designed for the data acquisition process. The mechanism is combined with a few stainless-steel parts as shown in Fig. 9.

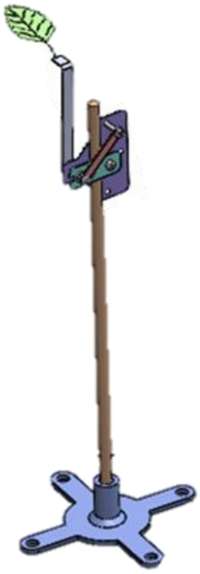

Fig. 9. Leaf holder mechanism

The design enables to connect the mechanism to the plate of the turntable, and to manually change 3 orientation options of the leaf attached to it: 0, 45, and 90 degrees, as shown in Fig. 10, Fig. 11 shows the actual leaf holder mechanism.

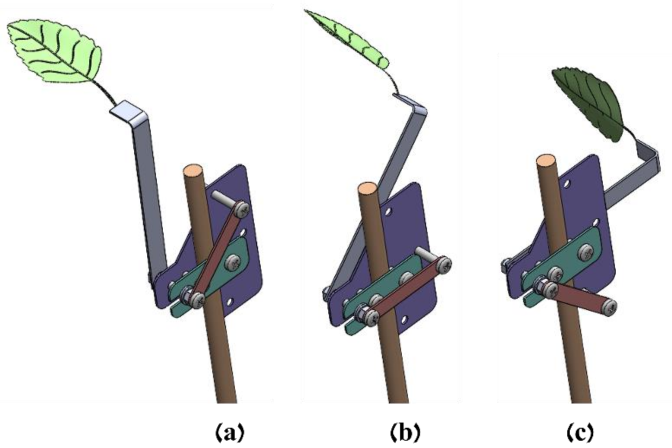

Fig. 10. Leaf primary orientation angle change - (a): orientation angle of 0°, (b): orientation angle of 45, (c): orientation angle of 90°

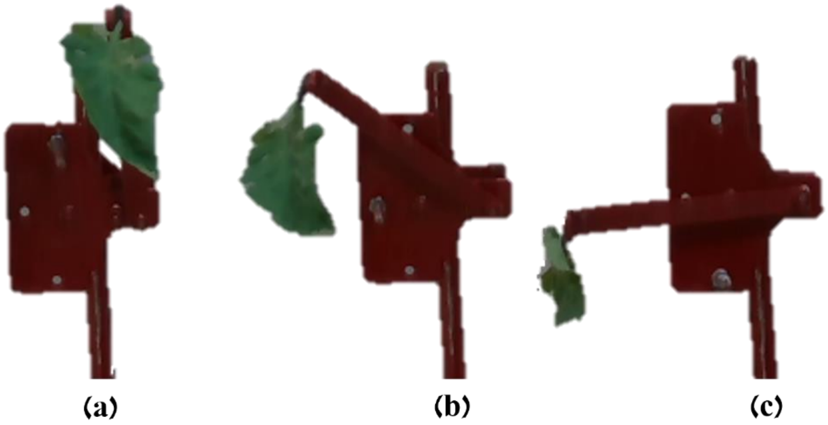

Fig. 11. Leaf holder real mechanism - (a): orientation angle of 0°, (b): orientation angle of 45, (c): orientation angle of 90°

During the data acquisition process, the UR5 robotic arm is planned to change the distance of the camera from the leaf's center, as well as the elevation angle of the end-effector in relation to the leaf center. The turntable is controlled by a step-motor connected to "Arduino" controller, and is rotating 360 degrees, in 30 degrees steps. The mechanism holding the tomato leaf enables to change the leaf primary orientation on the pole.

3.2.5 PC and Control Software

Integrating the experiment apparatus to an automatic system requires using a computer and designing a control software for the data acquisition. The software is written in python oversees moving the robotic arm (by communicating the UR5 controller), and the turntable (by communicating the Arduino controller), as well as taking images with the camera and saving all data.

Communication with the UR5 robotic arm controller is done by using "socket" python package that provide the ability to receive and send messages across the network, as both PC and UR5 controller are connected to. Socket functions are used to read the robotic current pose, as well as to send commands to the controller to move the robotic arm. This command is combined with 6 values sent- 3 values for x, y, z position, and 3 values for the rotation around x, y, z axis of the TCP desired coordinate system in relation to the coordinate system of the base of robot.

Controlling the turntable is done by "serial" python package that allows communicating over a serial port (that is used with the Arduino controller). The messages sent to the controller activate a rotation function in the Arduino controller that is connected to the turntable motor.

The communication with the camera is done by a "pyrealsense" python package that enables to open a pipeline to control the camera's parameters and streaming the camera view when needed. Combining the devices in one software allows synchronizing all parts of the experiments automatically, as well as saving layered data for every image containing additional information of both the robotic and turntable position.

3.3 Variables definitions

3.3.1 Coordinate Systems Definitions

In order to plan the robotic motion and calculate the equivalent labelled normal vector for each image, it is necessary to define the different coordinate systems in the experiment apparatus. Fig. 12, Fig. 13 and Fig. 14 present the different coordinate systems, and Table 1 include the definition of each one of them.

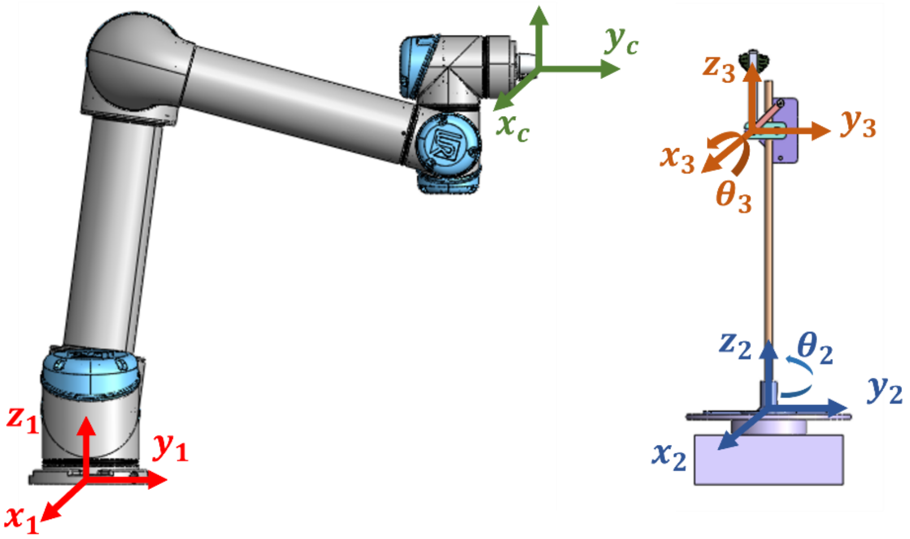

Fig. 12. Experiment Apparatus - Coordinate Systems 1,2,3,C of the robot base, camera and turntable, θ_2, θ_3 present the rotating angle of the turntable, and the rotation angle of the leaf holder.

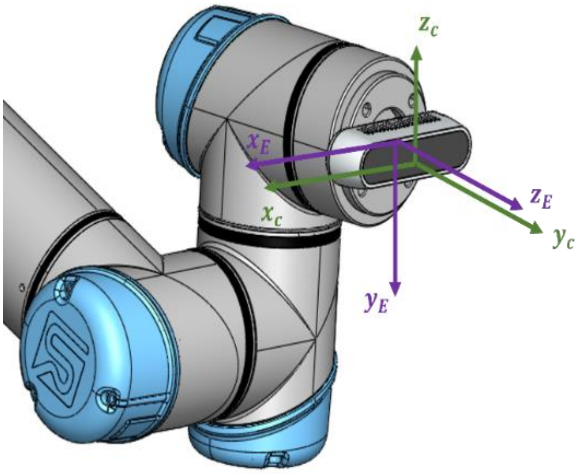

Fig. 13. Experiment Apparatus - Coordinate Systems: E of the Robot End Effector, and C of the Camera

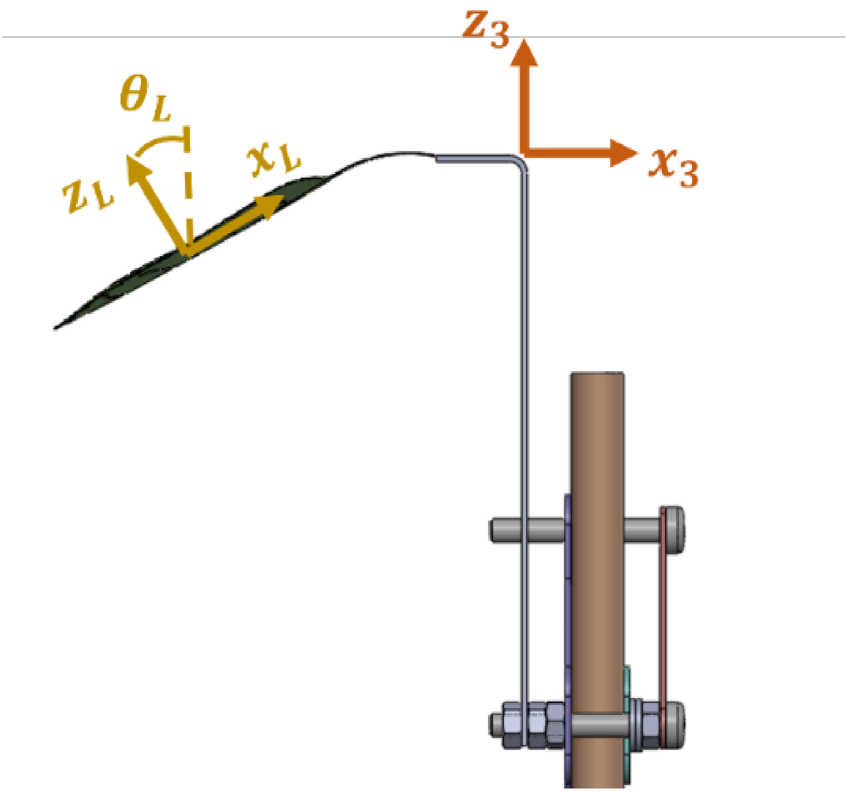

Fig. 14. Experiment Apparatus - Coordinate Systems 3, L: of the leaf holder and the leaf

Table 1. Coordinate Systems of the Experiment Apparatus

Coordinate System	Definition
1	UR5 base
2	Turntable
3	Leaf Holder
L	Leaf
E	End-effector
C	Camera

3.3.2 Experiment Variables

The independent variables in this experiment are the different leaves examined, the distance from the camera to the leaf center (as shown in Fig. 15), the rotation angle of the turntable θ_2, the robot arm elevation angle θ_E (Fig. 16) and the primary orientation angle of the leaf changed by the mechanism θ_3. The dependent variables in this experiment include the leaf orientation in relation to the camera- this orientation is defined to be the vector perpendicular to the leaf surface, $\overline{N_{Leaf}}$, and is dependent on the angles changed during the data-acquisition process, and the Image taken from the real-sense camera (the image changes according to the leaf's orientation). Table 2 shows the variables defined in the experiment.

Table 2. Angles of the Experiment Apparatus

Variables	Definition
θ_2	Rotation angle
θ_3	leaf holder rotation angle
θ_L	Leaf curvature angle
θ_E	Robot elevation angle
R	Distance from leaf center
$\overline{N_{Leaf}}$	Leaf normal vector

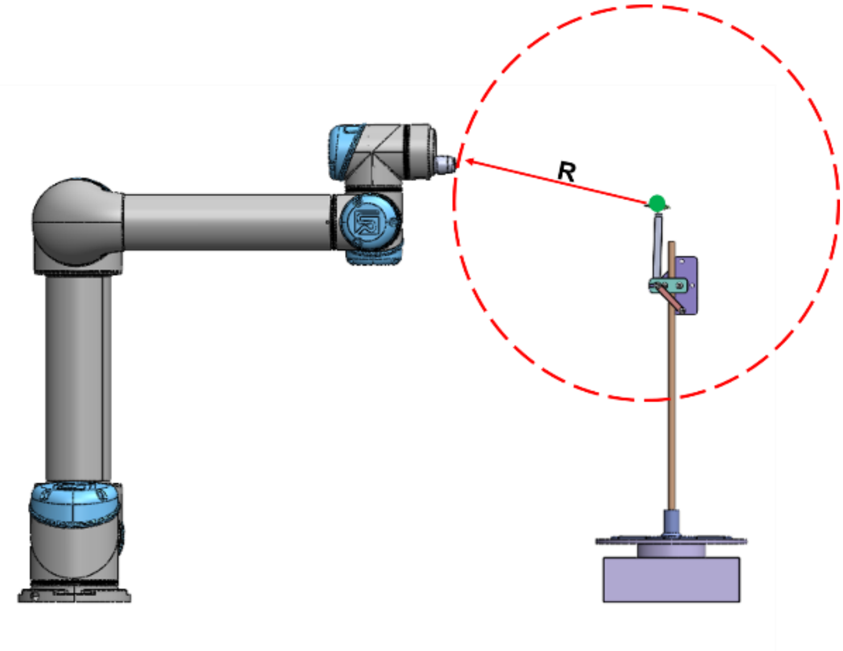

Fig. 15. Changed distance from leaf center

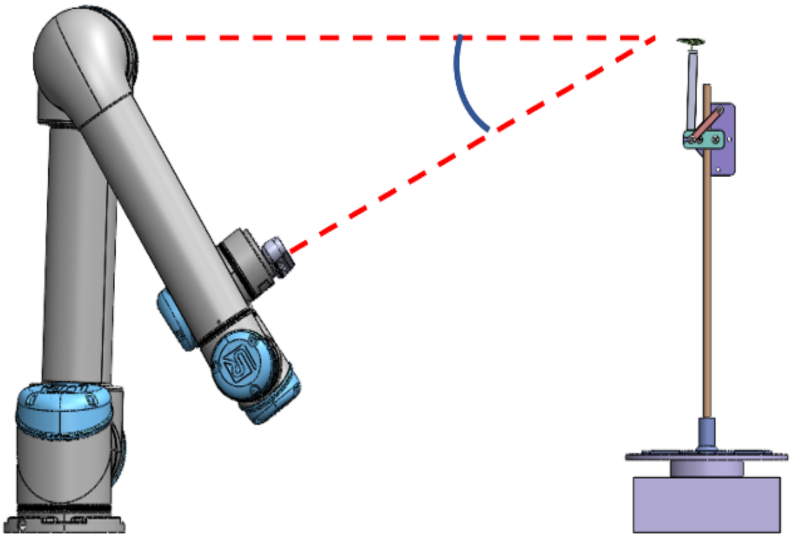

Fig. 16. Changed elevation angle of robotic arm

3.3.3 Transformation Matrices

Transformation matrices between the coordinate systems are calculated. Equation (1) shows the transformation metrics between the UR5 robot base coordinate system and the turntable- the transformation is the rotation of the turntable in rotation angle θ_2 around its Z axis.

$$T_{12} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Equation (2) show the transformation metrics between the turntable to the leaf holder mechanism- the transformation is the leaf holder rotation angle θ_3 around X axis.

$$T_{23} = \begin{bmatrix} 1 & 0 & 0 \\ \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 1 \end{bmatrix} \tag{2}$$

Equation (3) shows the transformation metrics between leaf holder mechanism to the leaf- the transformation is the angle depend on the curvature of the leaf as it is placed on the holder and is defined as θ_L around -Y axis.

$$T_{3L} = \begin{bmatrix} \cos(-\theta_L) & 0 & \sin(-\theta_L) \\ 0 & 1 & 0 \\ -\sin(-\theta_L) & 0 & \cos(-\theta_L) \end{bmatrix} \tag{3}$$

Equation (4) show the transformation metrics between the end effector to the UR5 robot base coordinate systems - the transformation is the elevation angle θ_E of the robotic arm, around X axis. Example of the elevation angle is shown in

$$T_{1E} = \begin{bmatrix} 1 & 0 & 0 \\ \cos\theta_E & -\sin\theta_E & 0 \\ \sin\theta_E & \cos\theta_E & 1 \end{bmatrix} \tag{4}$$

Then, the transformation metrics T_{E1} is calculated by the inverse as shown in equation (5):

$$T_{E1} = T_{1E}^{-1} \tag{5}$$

In addition, as shown in Fig. 13, the transformation between coordinate system of the end effector and the camera can be defined in equation (6).

$$T_{CE} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{6}$$

Therefor the transformation between the camera coordinate system and the leaf coordinate system can be calculated with equation (7):

$$T_{CL} = T_{CE} \times T_{E1} \times T_{12} \times T_{23} \times T_{3L} \tag{7}$$

3.3.4 Leaf Normal Vector Definition

By knowing θ_L, θ_2, θ_3, and θ_E, the orientation of the leaf relatively to the camera can be calculated at each step and position of the experiment apparatus. The normal vector perpendicular to the leaf surface is the vector $\overline{Z_L}$ as shown in Fig. 14, represented in the coordinate system of the camera (C). Equation (8) shows the calculation of this vector.

$$\overline{N_{Leaf}} = \overline{Z_L} = T_{CL} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{8}$$

3.3.5 Robotic Transform Data

The robotic end-effector position and orientation is known during the different steps of the experiment. The end-effector pose in image n is then defined in equation (9):

$$E_{p_n} = [x_n, y_n, z_n, R_{x_n}, R_{y_n}, R_{z_n}] \tag{9}$$

Where E_{p_n} is the end-effector pose vector for image n, x_n, y_n, z_n are the position of the end-effector when capturing image n and $R_{x_n}, R_{y_n}, R_{z_n}$ are the rotation of the end-effector, around each axis of the coordinate system of base of the UR5 robotic arm.

Using the robotic pose itself as input for the deep neural network can affect the generalization of the model and cause over-fitting, by "connecting" a certain end-effector pose to a specific orientation, where in "real-time" the same orientation can be detected from different poses of the end-effector. This is the reason to use the difference between poses - that represents the change of the positions (independent of the starting pose). Example of the calculation of the change of the robotic end-effector position is calculated in equation (10):

$$\Delta E_{p_{ik}} = E_{p_k} - E_{p_i} = [\Delta x_{ik}, \Delta y_{ik}, \Delta z_{ik}, \Delta R_{x_{ik}}, \Delta R_{y_{ik}}, \Delta R_{z_{ik}}] \tag{10}$$

Where $\Delta E_{p_{ik}}$ is the change of position and orientation of the robotic end-effector, E_{p_k} and E_{p_i} are the end-effector pose vectors for images k and i. Δx_{ik} is the change of the robotic end-effector position in x axis, similarly the Δy_{ik} is the change of the robotic end-effector position in y axis, and Δz_{ik} is the change of the robotic end-effector position in z axis.

$\Delta R_{x_{ik}}$ is the change of rotation angle of the robotic end-effector around the x axis, and similarly $\Delta R_{y_{ik}}$ is the change of rotation angle of the robotic end-effector around the y axis and $\Delta R_{z_{ik}}$ is the change of rotation angle of the robotic end-effector around the z axis.

3.4 Data Acquisition Process

For every leaf in this experiment, images in different orientations are collected and the vector perpendicular to the leaf surface, is calculated in the coordinate system of the camera.

3.4.1 Leaf Primary Orientation Calculation

The data acquisition process first stage is done by placing the leaf on the leaf holder in front of the camera. The camera initial pose is defined as the leaf y axis Y_L and the camera Z_L axes are coincident – which means the camera is straightened in front of the leaf, in the height of the leaf holder mechanism. An example for the primary positioning of the leaf as it captured with the camera is shown in Fig. 14. Leaf curvature angle θ_L is set for each leaf. When the leaf and the camera are orientated in the position described, the leaf curvature angle θ_L can be calculated directly from the RGB image. The calculation is done by extracting 20 points on the boundary of the leaf using the method detailed in 'Image processing' section (Fig. 17b), Then constructing a line that has the best fit to this set of points, by minimizing the squared error E defined in equation (11) this line represent the direction of the leaf. This step is shown in Fig. 17c, and Fig. 17d:

$$E = \sum_{i=0}^{20} \left| m_1(x_j) + b - y_j \right|^2 \tag{11}$$

Where E is the squared error, $x_j,\ y_j$ are the pixel j values in the set of boundary points of the leaf in x and y axis of the image, m_1 is the slope of the fitted line, and b is the point where this line crosses the y axis.

The direction perpendicular to the line with slope m_1 represent the leaf normal vector in the image. This normal vector slope is then calculated simply as shown in equation (12) (Fig. 17e):

$$m_2 = -\frac{1}{m_1} \tag{12}$$

Where m_2 is the slope of the perpendicular line.

Finally, the angle of the curvature of the leaf, θ_L, is the angle between the normal vector to the vertical axis of the camera and is calculated in equation (13). This step is shown in Fig. 17e:

$$\theta_L = \arctan(m_2) - 0.5\pi \tag{13}$$

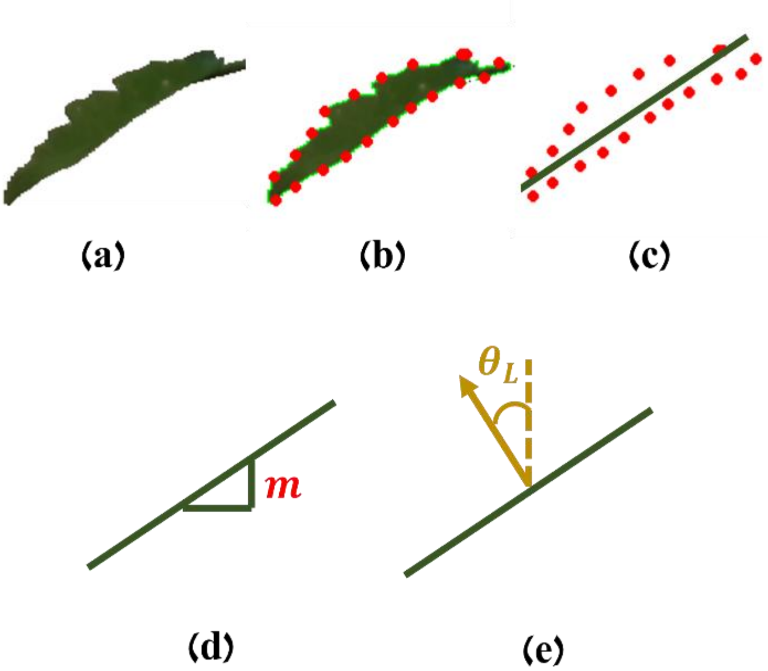

Fig. 17. Calculating the angle of the curvature of the leaf process: (a) Image sample (b) Boundary points detected (c) fitting line (d) Slope of the fitting line (e) Angle θ_L

3.4.2 Data Labelling

For every image the elevation angle of the robot θ_E, the rotation angle of the turntable θ_2 and the rotation angle of the leaf holder θ_3 are known as they are set in the process of the data acquisition. θ_L- the angle of the curvature of the leaf is calculated once for every leaf. Knowing those angles for every step of the data acquisition process, every image captured is labeled with the equivalent normal vector by using equation (8). This vector is the leaf normal vector perpendicular to the leaf surface represented in the camera coordinate system. This vector is a 3 X 1 form that will be the output for the deep neural network.

3.5 Data Processing

In the data Processing stage, the raw data collected - the images that were captured are processed to extract the equivalent leaf boundary points vector of every leaf. This is done by image processing methods, including isolating the leaf from the background, detecting the contour of the leaf and selecting 20 equally spaced points on the contour.

In addition, different datasets have been examined to include more data as input to solve the challenge of converting the 2-dimensional image data into a 3-dimensional estimation. Two main additions were tests: First, leaf boundary points vectors of more than one image were combined to make one input array. Second, the known change of the robotic end effector pose (the camera pose) between every pair of images is combined to the existing input array of those equivalent images.

3.5.1 Image processing

Every RGB image sample (Fig. 18a) is converted to HSV color space, and the green color (of the leaf) is being filtered using ”OpenCV” functions (Fig. 18b). Then, the boundary of the leaf is found (Fig. 18c), and 20 boundary points (x, y pixels) around the leaf are extracted (Fig. 18d)

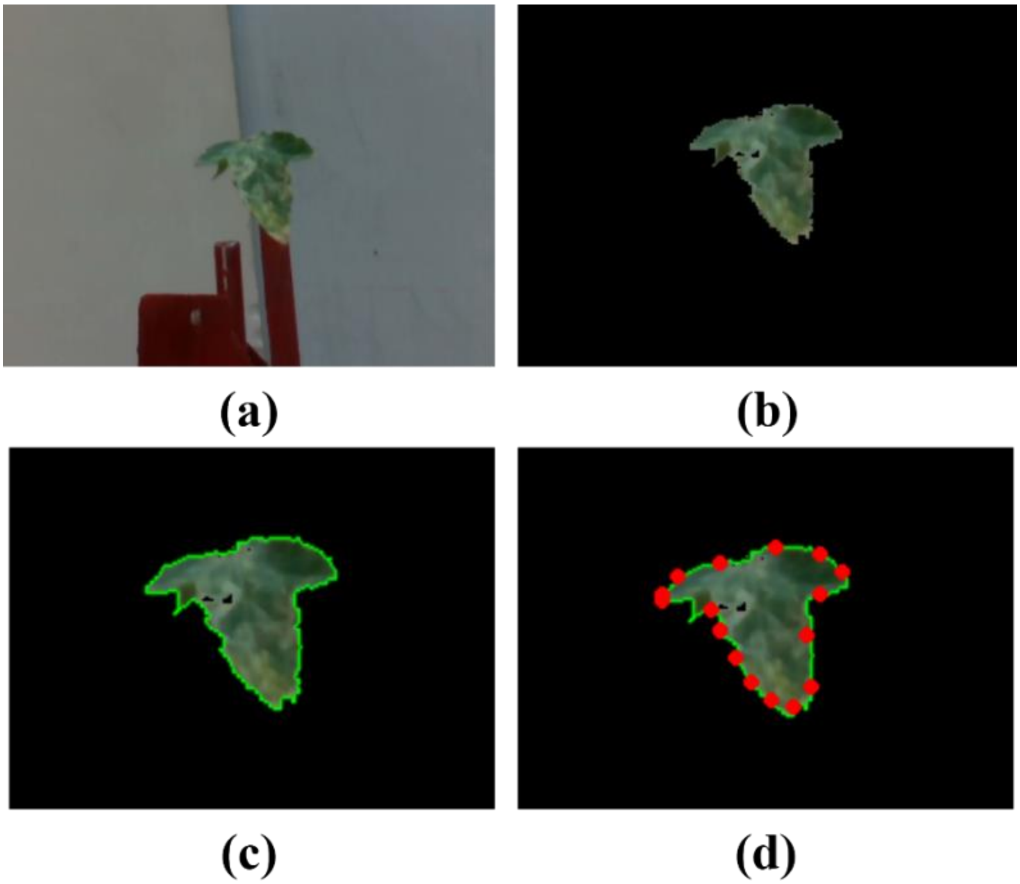

Fig. 18. Image processing: (a) Image sample (b) Green filtering (c) Boundary detection (d) Boundary points extraction

Using the leaf boundary points vectors as they appear in the image can affect the generalness of the model. Since the data acquisition process is repetitive, leaves with similar labeled orientation will appear in a certain area of the image, therefor it is possible for the model to associate the location of the leaf in the image as an indicator of the orientation of the leaf, which means the model would perform only for the images of leaves taken in the process of this data collection, and will not be reliable for determine the orientation for leaves in real time. To prevent this, the leaf boundary points vector values are shifted in both x, y axis. The size of the value used to shift every image for x and y is the minimum value of x pixel and the y pixel in the leaf boundary points vector, correspondingly. Fig. 19 demonstrate an example of the modification done for a sample of leaf boundary points vector. In this case the leaf was in the down right side of the image.

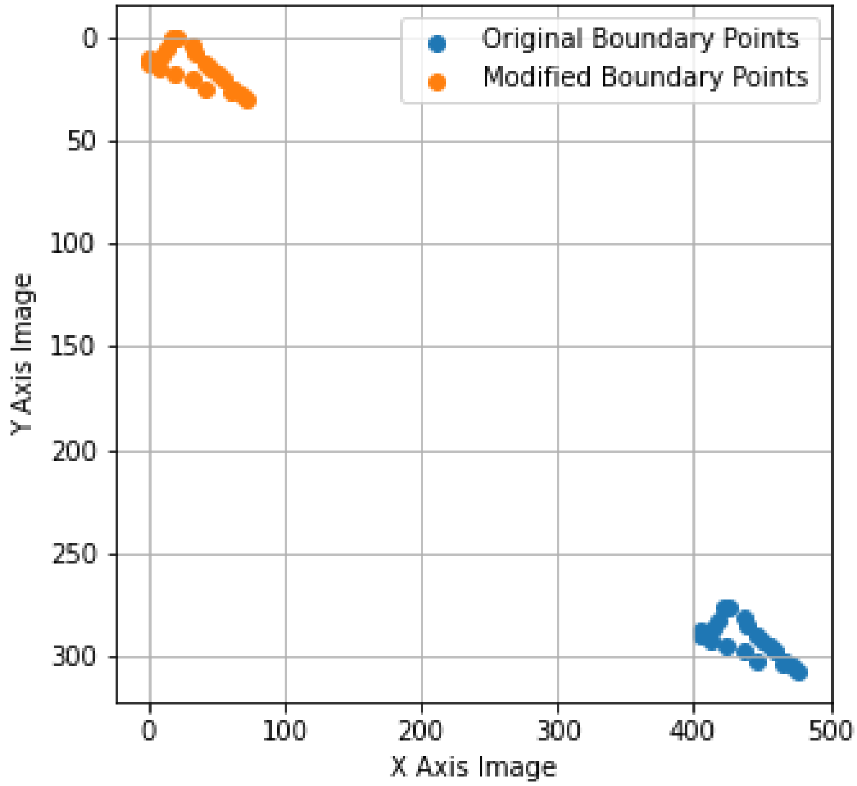

Fig. 19. Example of the modification for a single leaf boundary points vector

3.5.2 Different Datasets Preparation

After image processing, every image is "converted" to its equivalent leaf boundary points array. Combinations of those images leaf boundary points arrays define the five different datasets trained for five different models: "1 image", "2 image", "3 image", "2 images & RTD","3 images & RTD ". (RTD = Robotic Transform Data)

The first dataset is used for the simple "1 image" model. Every sample in this dataset contains a leaf boundary points vector as input for a single image.

The "2 images" and "3 images" datasets are used for the "2 images model" and "3 images model". Each sample in those datasets contains a combination of leaf boundary points vectors extracted from 2 or 3 images (respectively) of the same leaf, that were taken from different elevation angles or differen t distances of the robotic arm in relation to the leaf. Equations (14) ,(15) and (16) illustrate the input for 1 image, 2 image and 3 image models, correspondingly .

$$\mathrm{Input}_{1a} = [x_1, y_1, x_2, y_2 \dots x_{20}, y_{20}] \tag{14}$$

$$\mathrm{Input}_{2a} = [x_1, y_1, x_2, y_2 \dots x_{40}, y_{40}] \tag{15}$$

$$\mathrm{Input}_{3a} = [x_1, y_1, x_2, y_2 \dots x_{60}, y_{60}] \tag{16}$$

The "2 images & RTD","3 images & RTD" datasets are used for the "2 images & RTD model" and "3 images & RTD model". Every sample in these datasets contains the leaf boundary points vectors as described for "2 images" and "3 images" datasets, and the known robotic change of the end-effector position between the images is added to the end of the input array. This robotic change of end-effector pose is referred to as RTD (robotic transform data). The RTD includes the change of pose of the robotic end-effector position in x, y, z axis in relation to the robotic base coordinate system, and the change of the angle of rotation around the same x, y, z axis between every pair of images. Therefore, the data added to the 2 image models includes 6 values (the change between the first and second image) and the RTD for the 3-image model includes 12 values (the change between the first and second images, and the change between the second and third image). Equations (17) and (18), illustrate the input for, 2 image & RTD and 3 image & RTD models, correspondingly. Table 3 presents the size of the input array for each of the datasets.

$$\mathrm{Input}_{2b} = [x_1, y_1, x_2, y_2 \dots x_{40}, y_{40}, \left[\Delta E_{p_{12}}\right]) \tag{17}$$

$$\text{Input}_{3b} = [x_1, y_1, x_2, y_2 \dots x_{60}, y_{60}, \left[\Delta E_{p_{12}}\right], \left[\Delta E_{p_{23}}\right]\,] \tag{18}$$

Table 3. Input array length for every model

Model No.	Description	Leaf Boundary Points vector length	ΔE_p Vector length	Total Input length
1	1 image	40	-	40
2a	2 images	80	-	80
3a	3 images	120	-	120
2b	2 image & RTD	80	6	86
3b	3 image & RTD	120	12	132

* RTD = Robotic Transform Data

3.6 DL Model

3.6.1 Model modifications

The neural network model is based on the method used in Martinez et al. (2017). The method shows that converting ground truth 2D joint locations to 3D space is a task that can be solved with a remarkably low error rate. Thus, in an analogy to the tomato leaves, the 16 "joints" of the human body, which used as input to the network, have been altered to equally spaced points on the boundary of the leaf in the RGB image. Fig. 20 simulates the points on a leaf image used as input for the network.

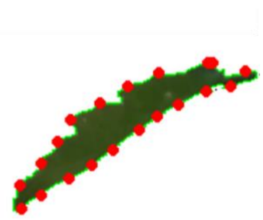

Fig. 20. Leaf boundary points

The output in the "3D human pose" model is a 16 by 3 array that represents the pose of all 16 human joints in the 3-dimentional space. Instead, in this research the equivalent output of the network is a spatial vector predicted as normal to the leaf surface (the position of the leaf is assumed to be known). This vector is a 3 by 1 vector, expressed in the robotic end effector coordinate system as illustrated in Fig. 21 .

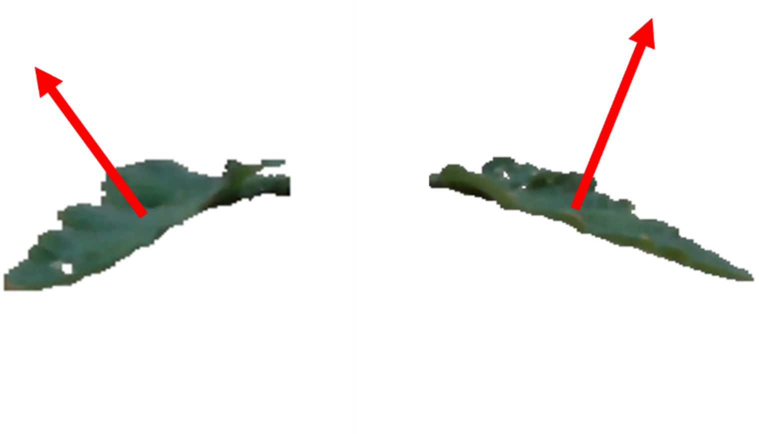

Fig. 21. Illustration of different leaves normal vector

The objective is estimating the most accurate normal vector, there for, the loss function of the "3D human pose" model should be altered. The value to minimize is the angle α – which is the angle between the ground truth- the labelled normal vector to the vector predicted by the network (Fig. 22). Meaning that the cosine value of the angle between the 2 vectors should be as close to the value 1 as possible. Then, the loss function is defined as shown in equation (19):

$$Loss = 1 - |\cos(\alpha)| \tag{19}$$

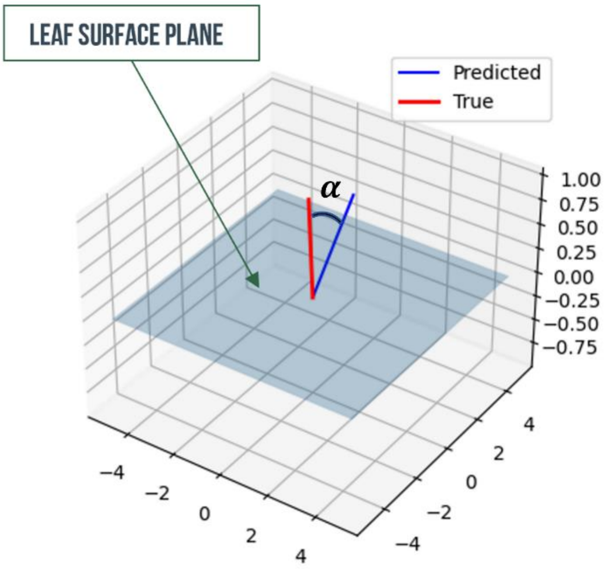

Fig. 22. Predicted and true vector and the angel α between them

The final architecture is a simple, deep, multi-layer neural network with batch normalization, dropout and Rectified Linear Units (RELUs), as well as residual connections. Fig. 23 shows the architecture of the network.

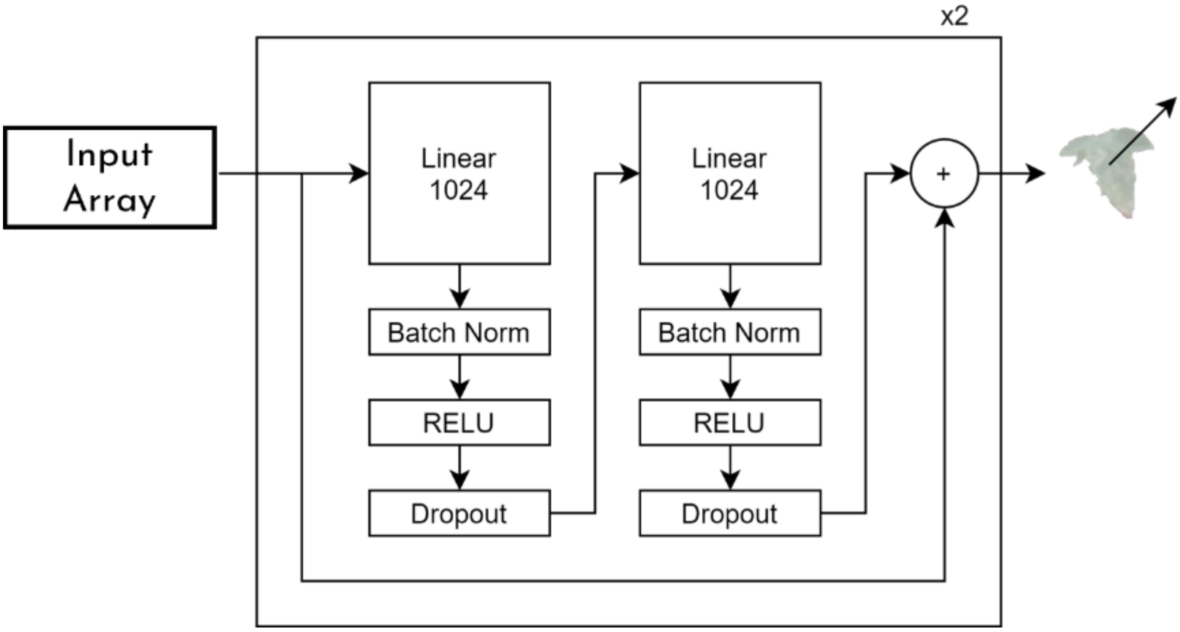

Fig. 23. Architecture of the deep neural network

3.6.2 Model Training

In this research, 5 different models are trained and evaluated. Table 4 shows the different types of models trained in this research, equivalent to the different types of datasets arranged.

Table 4. Different Model Types

Mode l No.	Description
1	1 image
2a	2 images
3a	3 images
2b	2 image & RTD
3b	3 image & RTD

* RTD = Robotic Transform Data

Each of the models are trained on optimizer "Adam" and different combination of hyper parameters (dropout, number of hidden layers, learning rate and batch size) to optimize the model performance. The hyper-parameters values are detailed in Table 5.

Table 5. Hyper- parameters changing values

Parameter	**Values**	**Possibilities**
Learning rate	[0.05,0.5,1]	3
dropout	[0,0.3]	2
No. hidden layers	[2,4,5]	3
Batch Size	[32,64]	2

3.7 Model Analyzing methods

The purpose of analyzing the model results is to improve the process of the model training, the "real time" operation, and the accuracy of the model estimation. Those purposes can be detailed into a few questions, as detailed:

3.7.1 Detection of the best performing model

This question will be answered by observing the straightforward results of the different models. The results will tell what combination of hyperparameters is producing the best accuracy. Also, the difference between the types of models depends on their datasets will be addressed to observe whether the use of 1,2, or 3 images as input changes the accuracy, and whether the addition of the RTD (robotic transform data) affect the results.

3.7.2 Quantifying the contribution of the additional images

To better analyze the contribution of the addition of images information as input, every set of images in the test dataset of "3 images & RTD" model is analyzed. Every sample of this dataset contains input from a set of 3 different images. The accuracy of the "3 images & RTD " on every set, is compared to the accuracy of the "2 images & RTD " model, given only the first 2 images in this set, and to the "1 image" model accuracy, given only the first image of this set. The comparison is done by calculating the change of accuracy (in degrees) between those three models.

3.7.3 Quantifying the contribution of the input values

Evaluation of the contribution of every value in the input array is important for analyzing what part of the input is more significant and effect the final estimation of the model, or less significant and can be removed from the input and shorten the computation time. The analysis is conducted for 3a,3b models.

The impact of the input values on the model estimation can be countified by calculating the gradient of the change of the input divided of the derived change of the model output. This is done by making a change of a single input value at a time and recalculating the output due to this change. This analysis is done by collecting the test dataset samples, and their original calculated outputs by the models. For each sample, single value of the input is changed by Δ, and the output is then recalculated.

Δ_i for every value indexed i of the input is determined by multiplication between a scale value s and the standard deviation of all the values in index i of the different samples in the test set. This way Δ_i is calculated by equation (20):

$$\Delta_i = 0.7 * \sigma_{samples[i]} \tag{20}$$

Δ_i is depended on $\sigma_{samples[i]}$ that is the standard deviation of all the values in the index i in the test set.

Equation (21) illustrates the original input and equation (22) illustrates the input after a change of a single variable in the index i, with the change of Δ_i.

$$I_{original} = [x_1, x_2, \dots, x_i \dots x_n] \tag{21}$$

Where $I_{original}$ is the original input structure.

$$I_{modified_i} = [x_1, x_2, \dots, x_i + \Delta_i, \dots x_{120}] \tag{22}$$

Where $I_{modified_i}$ is the modified input after an addition of Δ_i to the value in index i.

Let the original output equivalent to the original input $I_{original}$ be referred as $O_{original}$, and the output derived from the modified input referred as $O_{modified_i}$, as shown in equation (23) .

$$O_{modified_i} = model(I_{modified_i}) \tag{23}$$

Where $O_{modified_i}$ is a function of $I_{modified_i}$ given the trained model. The change, Δ_{output_i} between the original output and the modified output is calculated as the angle between 2 vectors, as shown in equation (24).

$$imp_i = \cos^{-1}\left(\frac{\overline{O_{modified_i}} * \overline{O_{original}}}{\left|O_{modified_i}\right| * \left|O_{original}\right|}\right) \tag{24}$$

imp_i represent the impact of the value i in the input array, on the model estimation. This is the angle of the change between the initial to the modified estimation, caused by a small change of this input value (as all the other values remain the same).

3.8 Experiments

3.8.1 Collecting Data

For all of the leaves, images are collected in different orientations: For every leaf, the leaf holder rotation angle θ_3 is changed (as shown in Fig. 10), for every angle θ_3, the turntable rotation angle θ_2 is changed, for every rotation angle, the robotic arm is changing the distance from the leaf (as shown in Fig. 15), and for every distance the elevation angle θ_E is changed (as shown in Fig. 16). The independent variables are arranged in

Table 6.

Table 6. Experiment independent variables

Variable	Values	Unit	No. Possibilities
Leaf number	[1:40]	-	40
Distance R	[0.3,0.5,0.7]	[m]	3
Rotation Angle θ_2	[0:360]	[°]	12 (steps of 30°)
Robot Elevation Angle θ_E	See Table 7		
Leaf Holder Rotation Angle θ_3	[0,45,90]	[°]	3

Number of steps and range of the robot elevation angle θ_E vary depends on the distance from the leaf due to the robotic arm limitations, Table 7 shows the change of the angle depends on the distance:

Table 7. Experiment Robot elevation angle θ_E

Distance R [m]	Values of robot elevation Angle θ_E [°]	No. Possibilities
0.7	[-30:30]	5 (steps of 15 °)
0.5	[-20:20]	3 (steps of 20 °)
0.3	[-10:10]	3 (steps of 10 °)

Images taken for every leaf are 396, in total 15,840 images were taken. Only images that were taken properly, or when the leaf is not hidden by the mechanism, are filtered from the raw database. After filtering the images, 10311 relevant images remained.

This procedure enables to collect a large database of labelled tomato leaf images with the spatial vector normal to the leaf surface. All independent variables are saved at each step of the experiment, and by changing one independent variable each step, different RGB images are taken by the" Real-Sense" camera, and the equivalent leaf normal is calculated in relation to the robotic end-effector. Fig. 23 shows an example of 3 different images of the same leaf at different orientations of the leaf taken during the data acquisition process.

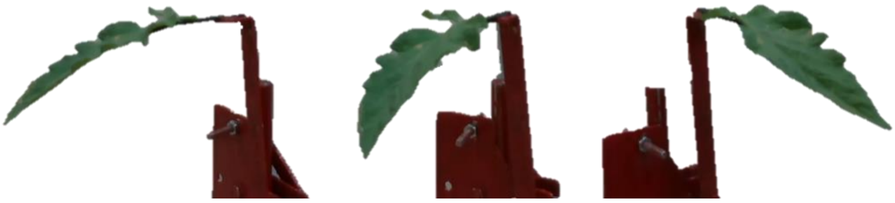

Fig. 24. Example of different images for the same leaf

Each of the images are processed and arranged in different dataset as described in 'Different Datasets Preparation' section, to prepare for the training step of the DL model.

3.8.2 Training the Models

Five different models were tested. Every model data set was divided to train, validation, and test sets; each set consists different leaves. After the process of filtering not-usable images, different datasets for the different models are created, as the size of every model dataset is changing. Table 8 shows the models trained and tested in the process of the experiment, along with the size of their equivalent datasets.

Table 8. Models trained in the experiment

Mode l No.	Description	Dataset size [No. Samples]			
		Train	Validation	Test	Total
	No. leaves	25	5	10	40
1	1 image	5269	1066	3976	10311
2a	2 images	44144	8600	33642	86386
3a	3 images	347094	65556	268116	680766
2b	2 image & RTD	44144	8600	33642	86386
3b	3 image & RTD	347094	65556	268116	680766

* RTD = Robotic Transform Data

In total, 180 different versions of the models were trained, validated and tested (36 versions for each model due to the change of hyper parameters). For every version, the epoch that yielded the best accuracy on the validation set, was saved as the best model of the training, each of the models was trained on 'NVIDIA A100 Tensor Core GPU' for 100 epochs at 'The Center for AI in Agriculture', Volcani Institute.

3.8.3 Evaluation

Evaluation of the model performance in real time is conducted by first determining the optimal routs for the robotic end effector to take the different leaf images to best predict the output (as detailed in the "Results" section), and secondly, by using a validation process of different new examples collected separately from the data collected for the model and in different distances to examine the actual performance of the model.

The validation procedure was conducted on 3 leaves (Fig. 25). Each leaf was placed on the leaf holder connected to the turntable. For every rotation angle of the turntable, the robotic arm moved from different starting points, in the optimal routs. The "ground truth" orientation of the leaf in relation to the camera was labeled. Using the set of images taken in this route, the orientation of the leaf was estimated by models 2a,2b,3a,3b, and the accuracy of the model estimation is calculated.

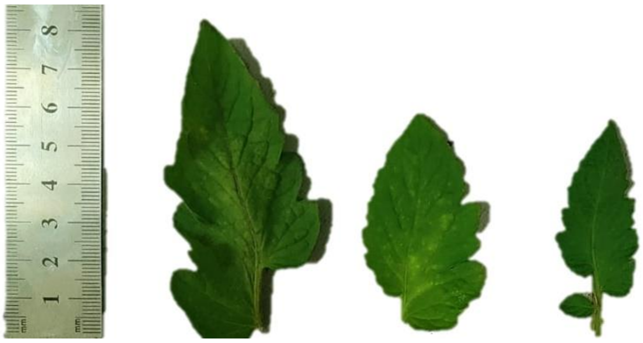

Fig. 25. Leaves tested in the validation process

Table 9 shows the number of different examples observed in the validation process and Fig. 26 shows example of one elevation angel of the robotic arm, selected in the process of evaluating the model.

Table 9. Validation examples collected

No. Optimal Routes	No. examples per route	No. rotation angles	No. Leaves	Total Examples
5	4	4	3	240

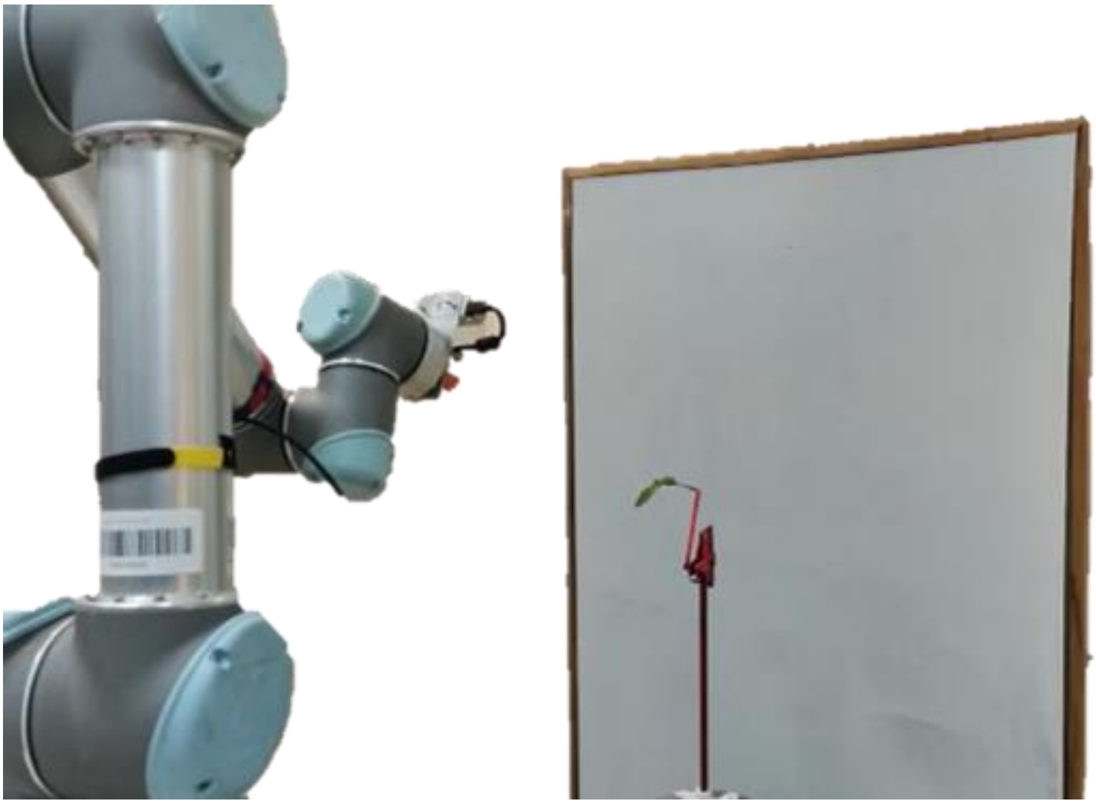

Fig. 26. Evaluation of the leaf orientation

The images taken by the camera from the robotic changing poses, are processed to use as input for the developed models. Models 3a, 3b (using 3 sets of images, with and without the data of the robotic transform) are used to estimate the normal vector of the leaf perpendicular to the leaf surface, in the camera coordinate system. The "ground truth" normal vectors are labelled in this process, so that the error of the estimation can be calculated.

4. Results

4.1 Model Main Results

Table 10 shows the best version of the models out of the 36 versions for each model. The table present the average accuracy for the train and test datasets, and the hyper-parameters on each one of the versions.

Table 10. Models main Results

Model No.	Description	Train Accuracy [°]	Test Accuracy [°]	Learning rate	Drop out	Hidden layers [#]	Batch Size
1	1 image	29.22	30.81	0.5	0	4	64
2a	2 images	17.56	19.16	0.05	0	5	64
2b	2 images & RTD	15.45	16.86	0.05	0	4	64
3a	3 images	11.45	14.28	0.05	0.3	2	32
3b	3 images & RTD	6.36	11.56	0.05	0	4	32

* RTD = Robotic Transform Data

The accuracy of "3 images & robotic transform" is 11.56°, this level of accuracy is satisfying to reach and grasp the leaf with a designated end effector. Furthermore, the "3 images" model (3a), is also sufficient for motion planning and can be used when robotic pose is not known or can't be used. Adding the information of the robotic transform improved the "2 images" model by 2.3° and the "3 images" model by 2.7°. This is a "simple" improvement that can be used in "real-time" when using the known robotic transform. Fig. 27 demonstrates orientation predictions of 3 different models (1,2,3 images models) for the same leaf.

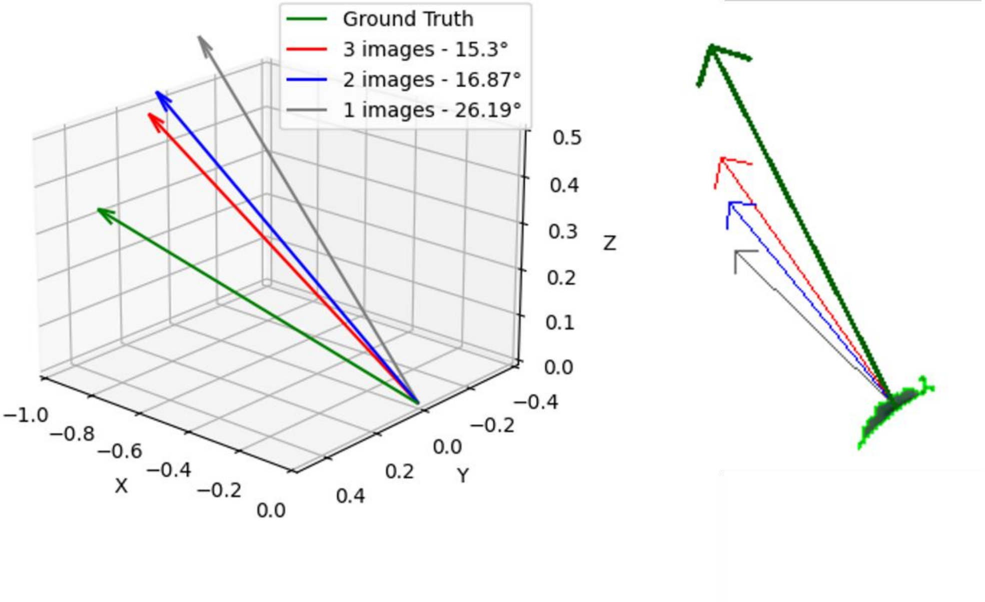

Fig. 27. Illustration of 3 different models' prediction

4.2 Input values contribution

The impact of model estimation of every input value i, is calculated as detailed in section 3.7.3 (value imp_i). This is a quantification on how much (in angles) the initial estimation is changing, due to a single small change of one of the input values (while all other values remain the same)

4.2.1 Leaf Boundary Points Vector Contribution.

Fig. 28 shows imp_i values for 40 values, that represents the leaf boundary points vector of 2 images (20 pixels for every image), for the 2 image models (2a, 2b). Each point on the graph represents how much the initial estimation of the leaf orientation has changed due to a small change of a single value of the input array.

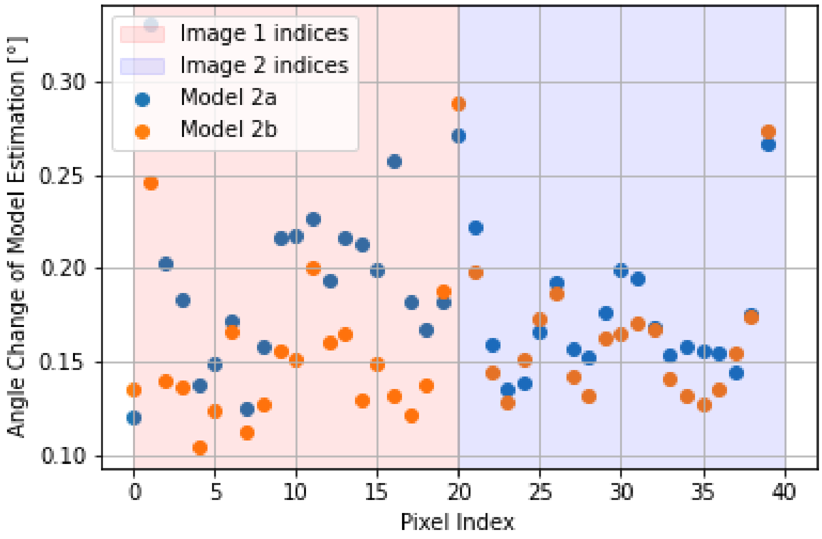

Fig. 28. Impact of the model estimation as a function of the different input values for models 2a,2b

Similarly, Fig. 29 shows impact of the 60 boundary points that represents the leaf boundary points vector of 3 images, for the 3 image models (3a,3b)

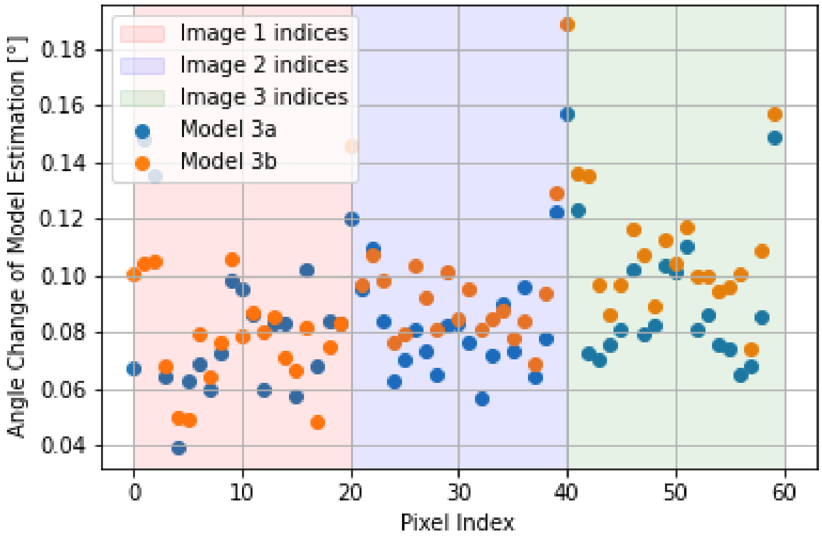

Fig. 29. Impact of the model estimation as a function of the different input values for models 3a,3b

Analyzing the input values contributions, a few conclusions can be made. First, the imapact of a single value of the boundary points vector is not significant if the rest of the values remains the same. The largest change is under 0.3[°] of the estimation. Yet, the relations between the different points impact can tell which of the boundary points is more important. Observing the graphs, the last and middle values of the leaf boundary points vector in every image seem to be the most important ones. Those indexes mostly indicate the pixel of the origin of the leaf (or close to it), and the pixel on the edge of the leaf in the opposite direction. This conclusion is important to prepare the image for the model in real time- it emphasis the importance of accurately detecting the origin and the edge of the leaf. The importance of the different pixels on the leaf boundary are illustrates in Fig. 30. The impact of every pixel in this figure is taken by the values computed for image 3 of model 3b.

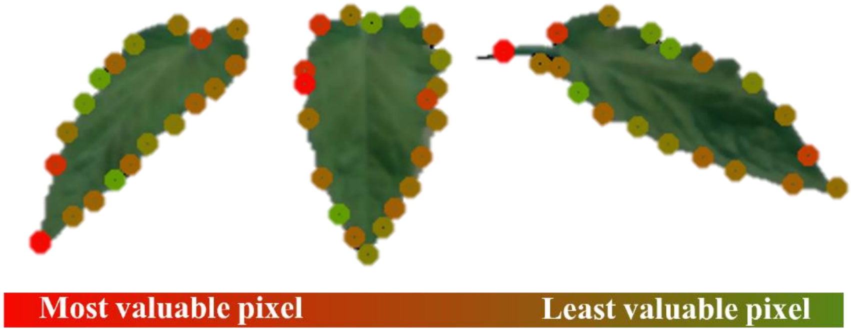

Fig. 30. Leaf boundary points importance in comparison to each other

4.2.2 Robotic Arm Transform Data Contribution.

From the results the contribution of the additional data is clear. The accuracy of the models using the RTD (robotic transform data) as input to the network is achieving better accuracy then their parallel models that only use the leaf boundary points vector of the different leaves as input, as the accuracy improves in approximately 3° in both models 2 and 3.

Additional analysis is performed to observe the contribution of the RTD in models 2b and 3b that are using this data as addition to the input. This analysis is done similarly to the analysis of the leaf boundary points vector analysis. The impact is calculated for the input values representing the end effector data. The graphs present the effect of every value on the prediction of the model- how a small change of this values can change (in angels) the initial prediction. This is done for models 2b, 3b that are using the RTD as addition to the leaf boundary points vector as input. Fig. 31 shows the effect of the RTD values on model 2b estimation. The percent values on Y axis refer to the change of the model prediction (in angles) caused by a small change of one value of the RTD.

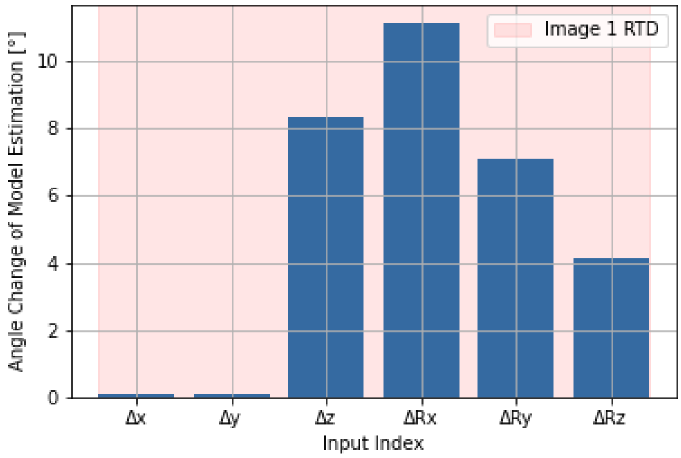

Fig. 31. Impact of the RTD values on the model estimation in model 2b.

Fig. 32 shows the effect of the RTD values on model 3b estimation.

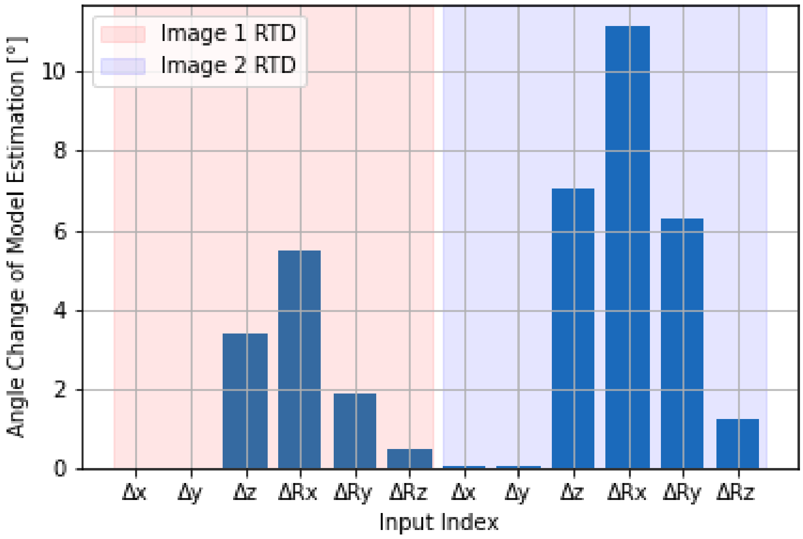

Fig. 32. Impact of the RTD values on the model estimation in model 3b.

The addition RTD values (Robotic transform data) for both models 2b,3b have great influence of the results. The changes of the position of the robot in x,y axis are insignificant since the robot position is only on the z axis. A change in the value of Z axis can change the model estimation in 5-7°. The changes of the rotation of the robot are highly important to the evaluation of the model since they hold the information about the elevation angle made from one image to the other. High values of the impact of model estimation shows that a small change of the values of the RTD, especial the values that hold information of the end effector orientation, have a great impact on the output of the model- up to 12° change from the initial estimation.

4.3 Additional Images Contribution

Fig. 34 shows the distribution of the change of accuracy between "1 image" model and "2 images & RTD" model, Fig. 37 shows the distribution of the change of accuracy between "2 images & RTD" model and "3 images & RTD" model, and Fig. 35 shows the distribution of the change of accuracy between "1 image" model and "3 images & RTD" model.

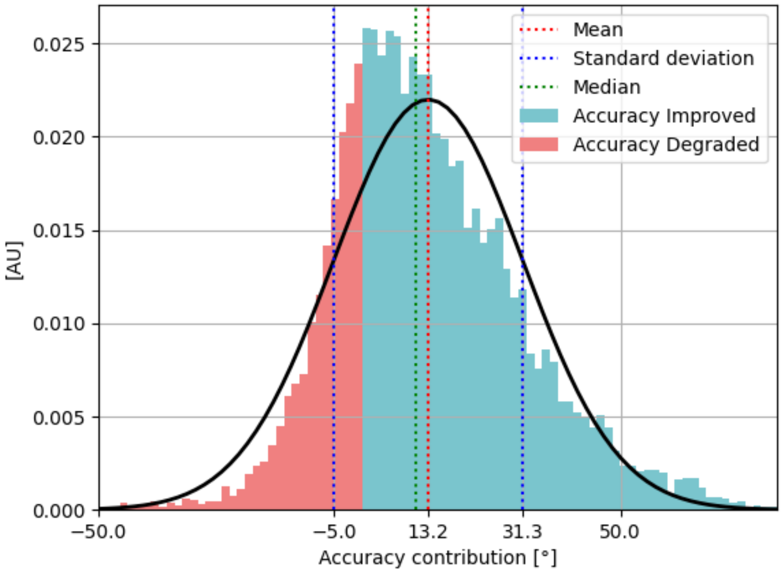

Fig. 33. Accuracy improvement between one image and 2 images predictions

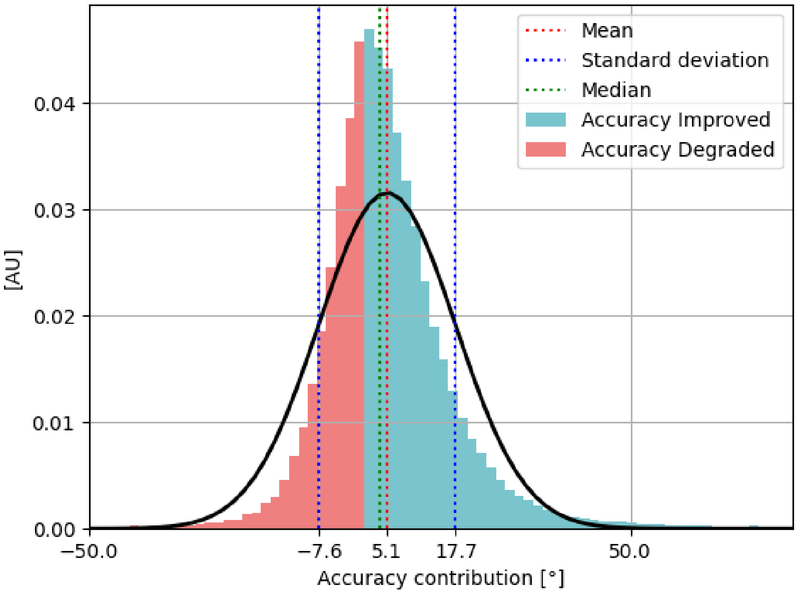

Fig. 34. Accuracy improvement between 2 images and 3 images predictions

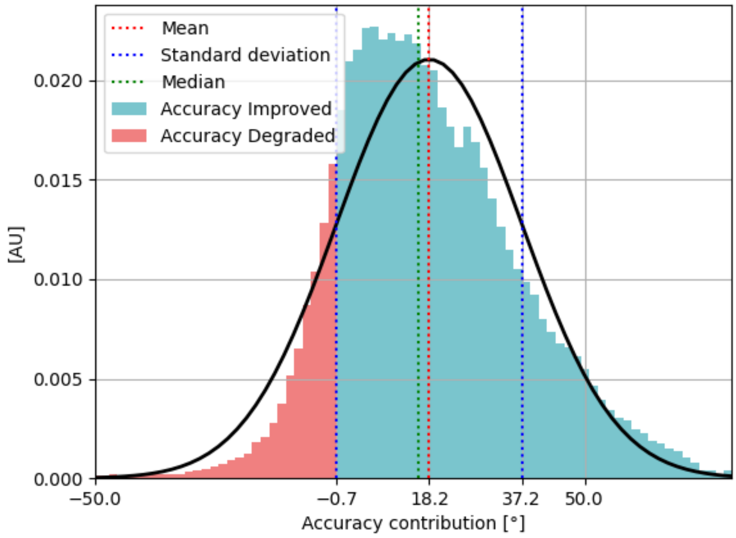

Fig. 35. Accuracy improvement between 1 image and 3 images predictions

Table 11 shows the probability to improve the accuracy by using additional images. This is the percent of cases improved by the addition of images by using the different models. Meaning the percent of the cases that are presented as the "Accuracy improved" bins in Fig. 33, Fig. 34, and Fig. 35 out of total number of cases.

Table 11. Additional Images Contribution

No. of existing images	No. of images added	Probability to improve accuracy [%]	Mean improvement [%]
1	+1	76.97	13.2
2	+1	67.63	5.1
1	+2	85.26	18.2

The results emphasis that adding images as input, significantly improves the model's performance. This is shown directly from Table 10 – presenting better accuracy for models using more images. In addition Fig. 33, Fig. 34, and Fig. 35 illustrates how every addition of image is improving the accuracy in most cases. Those distributions emphasis that the addition of a second image is more significant to the accuracy then adding a third image, as most of the cases of adding a second will improve the accuracy of the 1 image model by approximately 13.2° by using the "2 images" model (Fig. 33), when adding a third image will improve the accuracy of the "2 images" model by approximately 5.1° by using the "3 images" model (Fig. 34). Using 3 images instead of only one will clearly improve the accuracy the most – by average improvement of 18.2° to the accuracy as shown in Fig. 35. Using 3 images will most likely prevent "confusions" of the estimation as in most cases the contribution is positive- as the standard deviation is between -0.7° to 37.2° change of the accuracy (when negative signs present lowering the accuracy). These conclusions are important to make the estimation in "real time" – using the robotic arm to take several images from different angles will lead to a more accurate estimation of the spatial vector perpendicular to the leaf surface.

4.4 Robotic Optimal Elevation Angles

4.4.1 Cumulative probability

Models 2 and 3 (with and without robotic transform information) perform the estimation by using combination of 2 or 3 images respectively, as input to the network. Analyzing the model's best performance as a function of the robotic arm elevation angle can indicate the preferable motion for the robot in "real time".

To conduct this form of analysis, the test set accuracy results were observed for the different input of different images as a function of the elevation angles of the robot that were used to take them. First the test set was divided into groups of all the different robotic route options (including the different possibilities for the robotic elevation angles), and their equivalent accuracy results. Second, the cumulative probability graphs are plotted for every group – that represents what is the probability that an example in the group will achieve an accuracy score that is less or equal to x, when x is a variable changing from 0 to 90°.

Fig. 36 and Fig. 37 present for best groups (the optimal options for the robotic arm elevation angles) of models 2b and 3b, the percent of test data that its error is lower than the error angle in x-axis. Fig. 38 and Fig. 39 present similar graphs for models 2a and 3a.

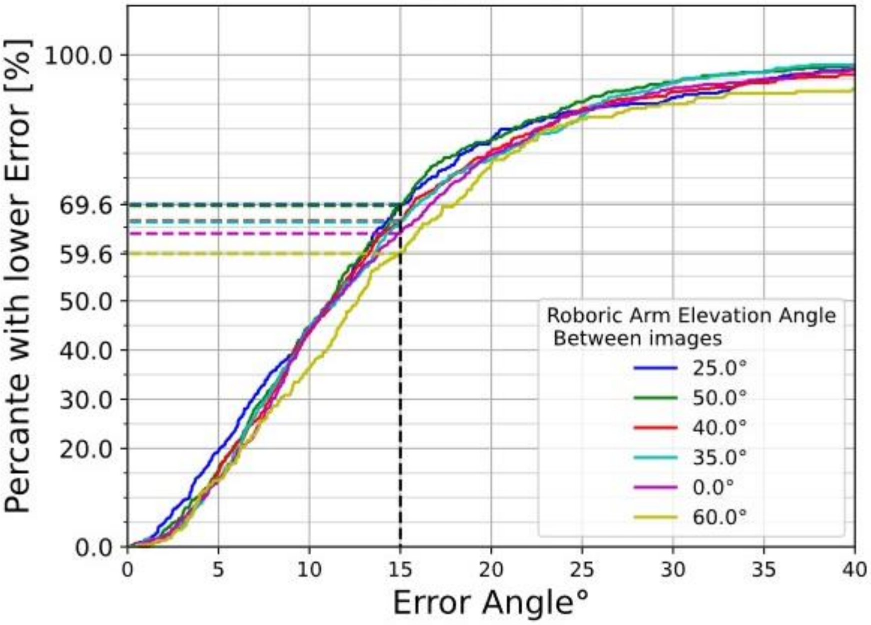

Fig. 36. Cumulative probability for different elevation angles in 2 images & robotic transform model (Model 2b)

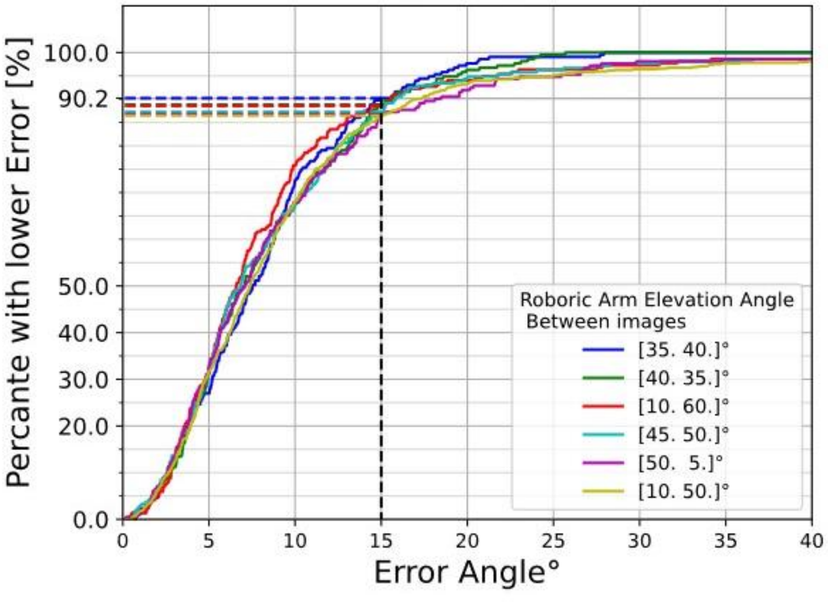

Fig. 37. Cumulative probability for different elevation angles in 3 images & robotic transform model (Model 3b)

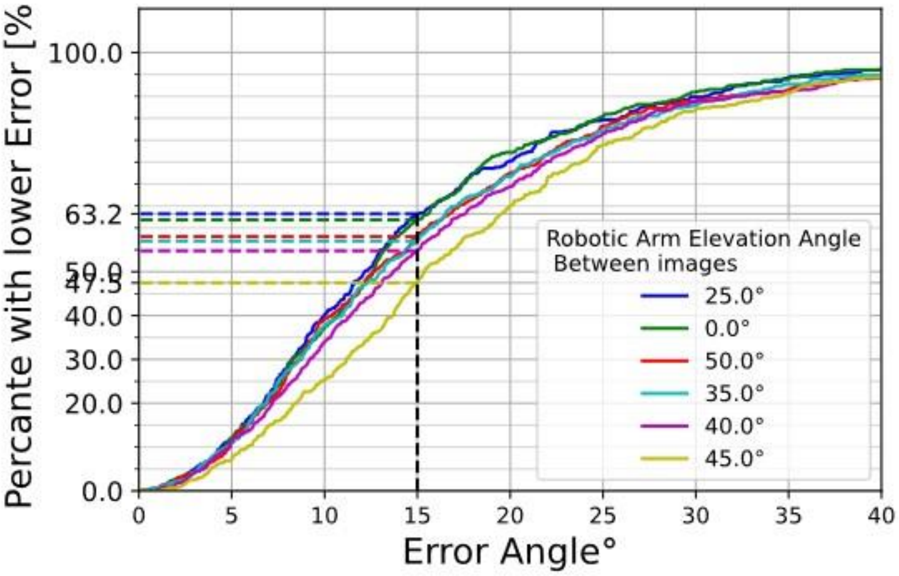

Fig. 38. Cumulative probability for different elevation angles in 2 images model (Model 2a)

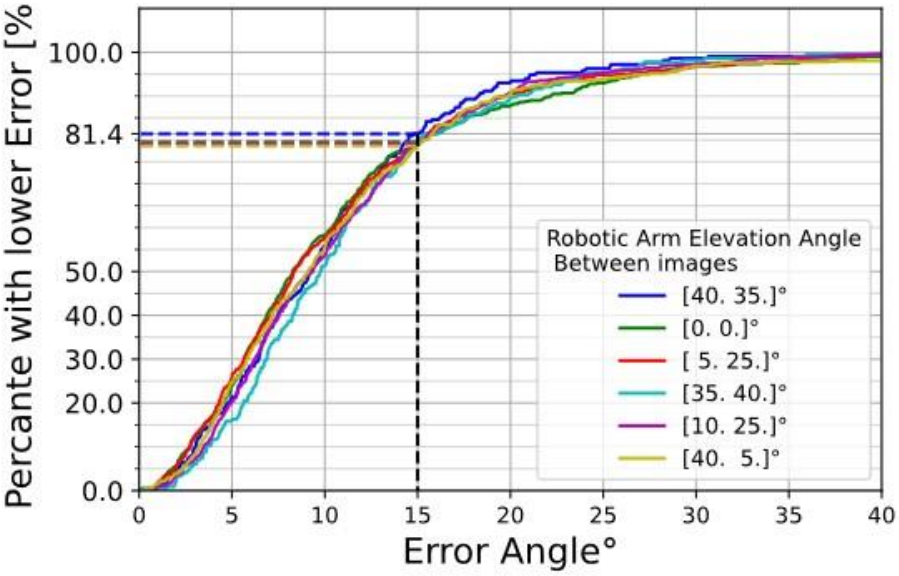

Fig. 39. Cumulative probability for different elevation angles in 3 images model (Model 3a)

By observing the cumulative probability graph for "2 image" model, the best elevation angle between the 2 images is 25° since it yields an error angle less than 15° for 69.6% of the test samples as shown in Fig. 36. For "3 image" model, the preferred elevation angles to take the 3 images are shown in Fig. 37 as they all yield the same accuracy of less than 15° for 90.2% of the test samples.

4.4.2 Selected Optimal Routs

From the cumulative probability analysis, the 5 best routes of the robotic arm have been selected, for both model 3a and 3b (3 images models). Table 12 and Table 13 shows the best elevation angles for those routes as well as their averaged accuracy score.

Table 12. Optimal routs of the robotic arm – model 3a (3 images)

No. Optimal Route	Elevation angles [°]	Average Error [°]
1a	[5, 30]	8.99
2a	[-5, 35]	9.24
3a	[-40, 5]	9.64
4a	[40, -35]	10.28
5a	[20, -35]	10.45

Table 13. Optimal routs of the robotic arm – model 3b (3 images & Robotic transform data)

No. Optimal Route	Elevation angles [°]	Average Error [°]
1b	[35, -40]	7.80
2b	[5, -40]	7.83
3b	[40,-5]	8.19
4b	[40, -35]	8.39
5b	[-35, -5]	8.45

4.5 Model Validation Results

In the validation process, the optimal routs were performed from different starting points to evaluate the leaves orientation in relation to the camera. Table 14 and Table 15 shows the accuracy of the different routes tested in the validation process for models 3a and 3b.

Table 14. Validation Process Results – model 3a

No. Optimal Route	Elevation angles [°]	Average Accuracy [°]			Weighted Average [°]
		Leaf 1	Leaf 2	Leaf 3	
1a	[5, 30]	20.40	24.37	23.89	22.89
2a	[-5, 35]	21.14	23.17	22.94	22.42
3a	[-40, 5]	24.37	31.46	17.47	24.43
4a	[40, -35]	21.55	42.64	12.87	25.69
5a	[20, -35]	20.58	23.67	21.17	21.81

Table 15. Validation Process Results – model 3b (with RTD)

No. Optimal Route	Elevation angles [°]	Average Accuracy [°]			Weighted Average [°]
		Leaf 1	Leaf 2	Leaf 3	
1b	[35, -40]	17.88	14.37	11.47	14.57
2b	[5, -40]	13.44	19.99	17.46	16.96
3b	[40, -5]	16.68	18.16	12.53	15.79
4b	[40, -35]	16.33	8.02	12.46	12.27
5b	[-35, -5]	11.30	12.09	20.22	14.54

4.6 Purposed Algorithm

An algorithm is built based on the results, to accomplish the "real time" task, from the primary stage of detecting the target leaf, to the stage of measuring the conductivity of the leaf and evaluating the plant's health. The algorithm is presented in the diagram in Fig. 40, including the steps after detecting the target leaf and its location and the process to estimate the leaf orientation.

The process starts with the detection of the leaf and evaluating its location relative to the camera mounted on the robotic arm. Then, a list of elevation angles is determined. The robotic arm then repeatedly changes the elevation angles around the known location of the leaf, takes and processes images of the leaf and saves the robotic end effector pose equivalent for every image. This process is done until reaching a specified number of valid images to use.

After all the data is collected using the camera and the robotic arm, it is processed to make an input for the trained model. The model receives the input and estimates the orientation of the leaf in a form of a 3 by 1 vector in relation to the robotic base coordinate system, representing the perpendicular vector to the leaf surface. Knowing the leaf orientation and the end effector orientation in relation to the robotic base coordinate system enables the calculation of the end effector desired orientation and position.

By this desired orientation, the robotic arm controller is planning the route for grasping the target leaf. After grasping the leaf, the measurement of the current is done using a designated gripper mounted on the robotic end effector, and by the current measurements, the health of the plant is evaluated.

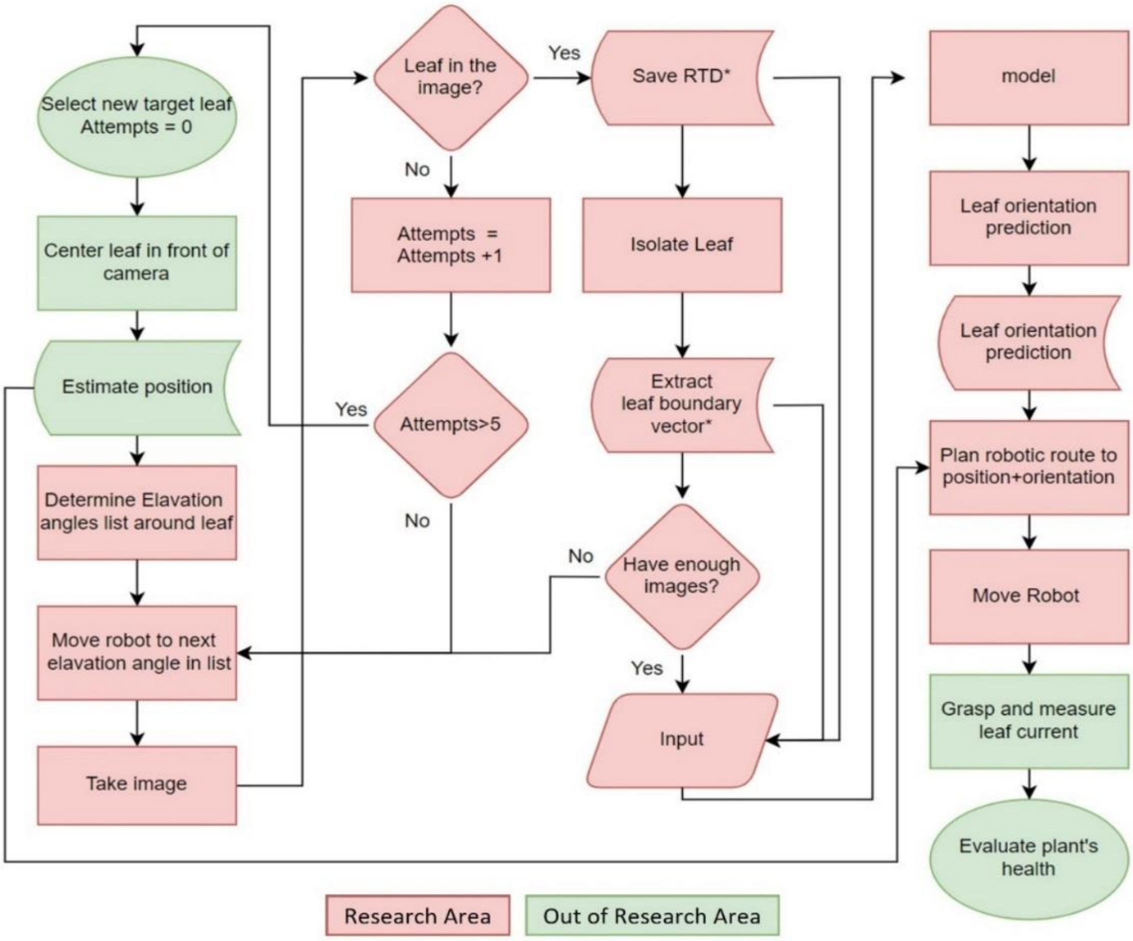
Select new target leaf
Attempts = 0
Center leaf in front of camera
Estimate position
Determine Elavation angles list around leaf
Move robot to next elavation angle in list
Take image
Leaf in the image?
Yes
No
Attempts = Attempts +1
Yes
Attempts>5
No
Save RTD*
Isolate Leaf
Extract leaf boundary vector*
No
Have enough images?
Yes
Input
model
Leaf orientation prediction
Leaf orientation prediction
Plan robotic route to position+orientation
Move Robot
Grasp and measure leaf current
Evaluate plant's health
Research Area
Out of Research Area

Fig. 40. Proposed solution algorithm

5. Discussion and Conclusion

Estimation of tomato leaves orientation is essential to reach and gently grasp the surface of the leaf task by a robotic arm to measure the leaf's resistance and assess development of a disease in early stages. The novel method developed in this research provide a direct solution to the task of leaf orientation estimation for manipulating a robotic arm. The research show that using a simple deep neural network trained on database of data from tomato leaves images, is sufficient to estimate the leaf pose, as the best model yield 11.56° average accuracy of the estimated spatial vector normal to the leaf surface and with errors lower than 15° for 90% of the cases. Furthermore, the data required from the images, can be reduced to a short array of numbers representing the leaf boundary points and allowing the simplicity of the network.

Adding known information about the robotic arm change of position as input to the network was proved to improve the accuracy of the estimation of the model. In addition, the accuracy of the models is significantly improved when the number of images used as input for the model is increasing. This way, by using leaf boundary points of 3 images yields the most accurate result (11.56°), comparing to the use of 2 images (16.86°) or one image (30.81°). Furthermore, the elevation angles of the robotic arm between images were proven to affect the accuracy as well, and optimal angles were investigated to plan the "real time" robotic motion to maximize the probability to achieve an accurate estimation.

The validation process of the model yields a few important conclusions and recommendations of operating the robotic arm in real time. First, it shows a better separation between the accuracy of the routes detected in the "optimal elevation angles" analysis and helps to determine the best routes to be made in "real time" by the robotic arm. In addition, the importance of the RTD information is again proven to significantly improve the accuracy of the estimation.

6. References

Ahlin, K., Joffe, B., Hu, A. P., McMurray, G., & Sadegh, N. (2016). Autonomous Leaf Picking Using Deep Learning and Visual-Servoing. *IFAC-PapersOnLine, 49*(16), 177–183. https://doi.org/10.1016/j.ifacol.2016.10.033

Alenyà, G., Dellen, B., Foix, S., & Torras, C. (n.d.). *Robotic Leaf Probing Via Segmentation of Range Data Into Surface Patches*.

Ampatzidis, Y., Bellis, L. de, & Luvisi, A. (2017). iPathology: Robotic applications and management of plants and plant diseases. In *Sustainability (Switzerland)* (Vol. 9, Issue 6). MDPI. https://doi.org/10.3390/su9061010

Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85–97. https://doi.org/10.1016/j.biosystemseng.2015.07.004

Barth, R., Hemming, J., & van Henten, E. J. (2016). Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, *146*, 71–84. https://doi.org/10.1016/j.biosystemseng.2015.12.001

Barth, R., Hemming, J., & van Henten, E. J. (2019). Angle estimation between plant parts for grasp optimisation in harvest robots. *Biosystems Engineering*, *183*, 26–46. https://doi.org/10.1016/j.biosystemseng.2019.04.006

Beadle, J., James Taylor, C., Ashworth, K., & Cheneler, D. (2020). Plant leaf position estimation with computer vision. *Sensors (Switzerland)*, *20*(20), 1–16. https://doi.org/10.3390/s20205933

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. In *Biosystems Engineering* (Vol. 149, pp. 94–111). Academic Press. https://doi.org/10.1016/j.biosystemseng.2016.06.014

Bechar, A., & Vigneault, C. (2017). Agricultural robots for field operations. Part 2: Operations and systems. *Biosystems Engineering*, *153*, 110–128. https://doi.org/10.1016/j.biosystemseng.2016.11.004

Bhaduri, B. L., Archibald, R., Bright, E. A., & Filippi, A. M. (2009). Hyperspectral agricultural mapping using Support Vector Machine-Based Endmember Extraction (SVM-BEE). *Optics Express, Vol. 17, Issue 26, Pp. 23823-23842*, *17*(26), 23823–23842. https://doi.org/10.1364/OE.17.023823

Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C. J., & Kumar, V. (2017). Counting Apples and Oranges with Deep Learning: A Data-Driven Approach. *IEEE Robotics and Automation Letters*, *2*(2), 781–788. https://doi.org/10.1109/LRA.2017.2651944

Daponte, P., de Vito, L., Glielmo, L., Iannelli, L., Liuzza, D., Picariello, F., & Silano, G. (2019). A review on the use of drones for precision agriculture. *IOP Conference Series: Earth and Environmental Science*, *275*(1), 012022. https://doi.org/10.1088/1755-1315/275/1/012022

Deepak, A. H., Gupta, A., Choudhary, M., & Meghana, S. (2019). Disease detection in tomato plants and remote monitoring of agricultural parameters. *Proceedings of the 11th International Conference on Advanced Computing, ICoAC 2019*, 28–33. https://doi.org/10.1109/ICoAC48765.2019.246812

Dharmaraj, V., & Vijayanand, C. (2018). Artificial Intelligence (AI) in Agriculture. *International Journal of Current Microbiology and Applied Sciences*, *7*(12), 2122–2128. https://doi.org/10.20546/ijcmas.2018.712.241

Durmu, H., Güne, O., & Krc, M. (n.d.). *Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning*.

Elavarasan, D., & Durairaj Vincent, P. M. (2020). Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications. *IEEE Access*, *8*, 86886–86901. https://doi.org/10.1109/ACCESS.2020.2992480

Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Santos, C. H., & Pekkeriet, E. (2020). Agricultural robotics for field operations. In *Sensors (Switzerland)* (Vol. 20, Issue 9). MDPI AG. https://doi.org/10.3390/s20092672

Fuentes, A., Park, D. S., Yoon, S., Youngki, H., & Lee, Y. (2016). *Characteristics of Tomato Plant Diseases-A study for tomato plant disease identification*. https://www.researchgate.net/publication/319952580

Hu, Y., Hugonot, J., Fua, P., & Salzmann, M. (2018). *Segmentation-driven 6D Object Pose Estimation*. http://arxiv.org/abs/1812.02541

Jeon, W. S., & Rhee, S. Y. (2017). Plant leaf recognition using a convolution neural network. *International Journal of Fuzzy Logic and Intelligent Systems*, *17*(1), 26–34. https://doi.org/10.5391/IJFIS.2017.17.1.26

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. In *Computers and Electronics in Agriculture* (Vol. 147, pp. 70–90). Elsevier B.V. https://doi.org/10.1016/j.compag.2018.02.016

Kenyon, L., Kumar, S., Tsai, W. S., & Hughes, J. d. A. (2014). Virus Diseases of Peppers (Capsicum spp.) and Their Control. In *Advances in Virus Research* (Vol. 90, pp. 297–354). Academic Press Inc. https://doi.org/10.1016/B978-0-12-801246-8.00006-8

Khan, R., & Debnath, R. (2019). *Segmentation of Single and Overlapping Leaves by Extracting Appropriate Contours*. 287–300. https://doi.org/10.5121/csit.2019.91323

Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, *14*(5), 778–782. https://doi.org/10.1109/LGRS.2017.2681128

Ling, K. S., Tian, T., Gurung, S., Salati, R., & Gilliard, A. (2019). First Report of Tomato Brown Rugose Fruit Virus Infecting Greenhouse Tomato in the United States. *Https://Doi.Org/10.1094/PDIS-11-18-1959-PDN*, *103*(6). https://doi.org/10.1094/PDIS-11-18-1959-PDN

Liu, J., & Wang, X. (2020). Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Frontiers in Plant Science*, *11*. https://doi.org/10.3389/fpls.2020.00898

Luo, L., Tang, Y., Zou, X., Ye, M., Feng, W., & Li, G. (2016). Vision-based extraction of spatial information in grape clusters for harvesting robots. *Biosystems Engineering*, *151*, 90–104. https://doi.org/10.1016/J.BIOSYSTEMSENG.2016.08.026

Luria, N., Smith, E., Reingold, V., Bekelman, I., Lapidot, M., Levin, I., Elad, N., Tam, Y., Sela, N., Abu-Ras, A., Ezra, N., Haberman, A., Yitzhak, L., Lachman, O., & Dombrovsky, A. (2017). A new israeli Tobamovirus isolate infects tomato plants harboring Tm-22 resistance genes. *PLoS ONE*, *12*(1). https://doi.org/10.1371/journal.pone.0170429

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A Simple Yet Effective Baseline for 3d Human Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*. https://doi.org/10.1109/ICCV.2017.288

Mishra, P., Polder, G., & Vilfan, N. (2020). Close Range Spectral Imaging for Disease Detection in Plants Using Autonomous Platforms: a Review on Recent Studies. *Current Robotics Reports*, *1*(2), 43–48. https://doi.org/10.1007/s43154-020-00004-7

Mohan, P., & Patil, K. K. (2018). Deep learning based weighted SOM to forecast weather and crop prediction for agriculture application. *International Journal of Intelligent Engineering and Systems*, *11*(4), 167–176. https://doi.org/10.22266/ijies2018.0831.17

Nguyen, T. T., Hoang, T. D., Pham, M. T., Vu, T. T., Nguyen, T. H., Huynh, Q. T., & Jo, J. (2020). Monitoring agriculture areas with satellite images and deep learning. *Applied Soft Computing*, *95*, 106565. https://doi.org/10.1016/J.ASOC.2020.106565

Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Tona, E., Hočevar, M., Baur, J., Pfaff, J., Schütz, C., & Ulbrich, H. (2016). Selective spraying of grapevines for disease control using a modular agricultural robot. *Biosystems Engineering*, *146*, 203–215. https://doi.org/10.1016/j.biosystemseng.2015.12.004

Oliveira, L. F. P., Moreira, A. P., & Silva, M. F. (2021). Advances in agriculture robotics: A state-of-the-art review and challenges ahead. In *Robotics* (Vol. 10, Issue 2). MDPI AG. https://doi.org/10.3390/robotics10020052

Palaniswami, C., Gopalasundaram, P., & Bhaskaran, A. (2011). Application of GPS and GIS in Sugarcane Agriculture. *Sugar Tech 2011 13:4*, *13*(4), 360–365. https://doi.org/10.1007/S12355-011-0098-9

Palazzi, V., Gelati, F., Vaglioni, U., Alimenti, F., Mezzanotte, P., & Roselli, L. (2019). Leaf-Compatible Autonomous RFID-Based Wireless Temperature Sensors for Precision Agriculture. *2019 IEEE Topical Conference on Wireless Sensors and Sensor Networks, WiSNet 2019*. https://doi.org/10.1109/WISNET.2019.8711808

Park, K., Patten, T., & Vincze, M. (2019). *Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation*. https://doi.org/10.1109/ICCV.2019.00776

Puri, V., Nayyar, A., & Raja, L. (2017). Agriculture drones: A modern breakthrough in precision agriculture. *Https://Doi.Org/10.1080/09720510.2017.1395171*, *20*(4), 507–518. https://doi.org/10.1080/09720510.2017.1395171

Reddy, S., Ben-Yashar, G., Dombrovsky, A., Jahn, Y., Ben-Shimon, Y., Bechar, A., & Yaakobovitz, A. (2022). Early Sensing of Tomato Brown Rugose Fruit Virus in Tomato Plants via Electrical Measurements. *IEEE Sensors Letters*, 1–1. https://doi.org/10.1109/lsens.2022.3161595

Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases. In *Computers and Electronics in Agriculture* (Vol. 72, Issue 1, pp. 1–13). https://doi.org/10.1016/j.compag.2010.02.007

Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., Imanol, ·, Muntión, L., Pape, J.-M., Polder, G., Vukadinovic, D., Yin, X., Tsaftaris, S. A., Scharr, H., Minervini, M., French, A. P., Muntión, · I, Klukas, C., Pape, J.-M., … Tsaftaris, S. A. (n.d.). *Leaf segmentation in plant phenotyping: A collation study*. http://www.plant-phenotyping.org/CVPPP2014

Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., & Berman, S. (2016). Robotic Disease Detection in Greenhouses: Combined Detection of Powdery Mildew and Tomato Spotted Wilt Virus. *IEEE Robotics and Automation Letters*, *1*(1), 354–360. https://doi.org/10.1109/LRA.2016.2518214

Schor, N., Berman, S., Dombrovsky, A., Elad, Y., Ignat, T., & Bechar, A. (2017). Development of a robotic detection system for greenhouse pepper plant diseases. *Precision Agriculture*, *18*(3), 394–409. https://doi.org/10.1007/s11119-017-9503-z

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience*, *2016*. https://doi.org/10.1155/2016/3289801

Steen, K. A., Christiansen, P., Karstoft, H., & Jørgensen, R. N. (2016). Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture. *Journal of Imaging 2016, Vol. 2, Page 6*, *2*(1), 6. https://doi.org/10.3390/JIMAGING2010006

Tellaeche, A., Burgosartizzu, X. P., Pajares, G., & Ribeiro, A. (2007). A vision-based hybrid classifier for weeds detection in precision agriculture through the bayesian and fuzzy k-Means paradigms. *Advances in Soft Computing*, *44*, 72–79. https://doi.org/10.1007/978-3-540-74972-1_11/COVER

Tomato brown rugose fruit virus (ToBRFV): A new concern for tomato and pepper producers - Vegetables. (n.d.). Retrieved October 12, 2022, from https://www.canr.msu.edu/news/tobrfv-a-new-concern-for-tomato-and-pepper-producers

Tomato brown rugose fruit virus (TOBRFV)[World distribution]| EPPO Global Database. (n.d.). Retrieved October 12, 2022, from https://gd.eppo.int/taxon/TOBRFV/distribution

Tsouros, D. C., Bibi, S., & Sarigiannidis, P. G. (2019). A Review on UAV-Based Applications for Precision Agriculture. *Information 2019, Vol. 10, Page 349*, *10*(11), 349. https://doi.org/10.3390/INFO10110349

van Henten, E. J., van Tuijl, B. A. J., Hemming, J., Kornet, J. G., Bontsema, J., & van Os, E. A. (2003). Field Test of an Autonomous Cucumber Picking Robot. *Biosystems Engineering*, *86*(3), 305–313. https://doi.org/10.1016/j.biosystemseng.2003.08.002

Viana, L. D. A., Tomaz, D. C., Martins, R. N., Rosas, J. T. F., Santos, F. F. L. dos, & Portes, M. F. (2019). Optical Sensors for Precision Agriculture: An Outlook. *Journal of Experimental Agriculture International*, 1–9. https://doi.org/10.9734/jeai/2019/v35i230203

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., & Guibas, L. J. (2019). *Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation*. http://arxiv.org/abs/1901.02970

Xiang, Y., Schmidt, T., Narayanan, V., & Fox, D. (2017). *PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes*. http://arxiv.org/abs/1711.00199

Yin, H., Cao, Y., Marelli, B., Zeng, X., Mason, A. J., & Cao, C. (2021). Smart Agriculture Systems: Soil Sensors and Plant Wearables for Smart and Precision Agriculture (Adv. Mater. 20/2021). *Advanced Materials*, *33*(20), 2170156. https://doi.org/10.1002/ADMA.202170156

Zappel, M., Bultmann, S., & Behnke, S. (2021). *6D Object Pose Estimation using Keypoints and Part Affinity Fields*. http://arxiv.org/abs/2107.02057

Zeng, A., Yu, K. T., Song, S., Suo, D., Walker, E., Rodriguez, A., & Xiao, J. (2017). Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge. *Proceedings - IEEE International Conference on Robotics and Automation*, 1386–1393. https://doi.org/10.1109/ICRA.2017.7989165

Zhang, Q. (n.d.). *PRECISION AGRICULTURE TECHNOLOGY FOR CROP FARMING*.

Zhang, S., Griffiths, J. S., Marchand, G., Bernards, M. A., & Wang, A. (2022). Tomato brown rugose fruit virus: An emerging and rapidly spreading plant RNA virus that threatens tomato production worldwide. *Molecular Plant Pathology*, *23*(9), 1262–1277. https://doi.org/10.1111/MPP.13229

Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., Jiang, Y., & Guo, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, *11*(4), 21–28. https://doi.org/10.25165/j.ijabe.20181104.4475

תקציר

גילוי מוקדם של מחלות בצמח העגבנייה הוא הליך חיוני למניעת הדבקה של פירות וחלקיו הנוספים של הצמח, כמו גם למניעה של התפשטות המחלה ליתר הצמחים השכנים ולמזעור אובדן היבול ופגיעה באיכות הפירות. מערכת רובוטית המצוידת בחיישן למדידת התנגדות חשמלית, פותחה על מנת לבצע את התהליך של הגעה, תפיסה ומדידת ההתנגדות החשמלית של עלה בצמח העגבנייה. ההתנגדות החשמלית התגלתה כמהווה אינדיקציה משמעותית המעידה על מצבו הבריאותי של הצמח בשלבים מוקדמים של המחלה. על מנת לבצע את פעולת תפיסת העלה ומדידת ההתנגדות באמצעות החיישן והזרוע הרובוטית, פותחה שיטה לחיזוי אוטומטי של אוריינטציית העלה תוך שימוש במאפיינים הנלקחים מתמונות שונות של עלים כקלט למודל למידה עמוקה.

לשם אימון המודל נערך ניסוי לאיסוף בסיס מידע באמצעות שימוש בזרוע הרובוטית UR5 במסגרתו צולמו ותויגו תמונות של עלי עגבנייה בכיוונים ובמרחקים שונים. כל אחת מהתמונות עובדה כדי לבצע בידוד של העלה המצולם מהרקע, וייצוא מערך המכיל רשימה של נקודות הממוקמות על היקפו של העלה בכל תמונה. כמו כן, מידע על המיקום והפוזיציה של הזרוע הרובוטית התווסף למערך זה. כל זאת על מנת להכין קלט עבור המודל לחיזוי האוריינטציה המדויקת של העלה הנבחן. המודל, לאחר קבלת הקלט, מאומן לייצא וקטור מרחבי המייצג את הווקטור המאונך לפני השטח של העלה. כדי לדייק את אוריינטציית העלה, אומנו מודלים שונים על קלט המכיל מידע עבור תמונה אחת, שתיים או שלוש של אותו עלה באוריינטציות שונות, בתוספת השינוי בין הפוזיציות השונות של המניפולטור בצילום של אותן תמונות.

התוצאות העיקריות מראות דיוק ממוצע של 11.5 מעלות בין הווקטור האמיתי, המאונך לפני השטח של העלה, לזה שנחזה על ידי המודל המאומן, תוצאה זו מספיקה על מנת לבצע את פעולת הרובוט לתפיסת העלה ומדידת הזרם בו באמצעות החיישן. בנוסף, תוצאות המודל נותחו כדי לייעל את זוויות איסוף התמונות האופטימליות לשם החיזוי המדויק ביותר של אוריינטציית העלה ושיפור הדיוק.

אוניברסיטת תל-אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן

בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

הערכת אוריינטציה של עלי עגבנייה באמצעות למידה עמוקה לשם זיהוי מוקדם של מחלות

חיבור זה הוגש כעבודת מחקר לקראת התואר ״מוסמך אוניברסיטה״ בהנדסה מכנית
על ידי

עדי כהן

העבודה נעשתה בבית הספר להנדסה מכנית

בהנחיית פרופ׳ אביטל בכר וד״ר אבישי סינטוב

תשפ״ג חשוון