BRAUDE
College of Engineering, karmiel

המכללה האקדמית להנדסה בכרמיאל
בראודה

Capstone Project Phase A

Crop Ripeness Identification by Analysis of Drone Video Stream

23-1-D-22

Supervisors: Dr Dan Lemberg, Mrs. Elena Kramer

Ori Malka mailto:Er.ori.malka@gmail.com

Alexander Martinov mailto:Alexmaartinov@gmail.com

Table of Contents

Abstract

The importance of crop farming in today's world, and throughout history cannot be overstated, as it can be considered the great initiator of civilization, and to this day provides one of the necessities for life. In recent years, precision agriculture has become an important area of research, with a focus on improving crop management and yield. One of the key challenges in precision agriculture is monitoring crop ripeness and identifying the number of fruits. In this project, we propose a crop management system using a drone to capture real-time video of citrus crops, and then use video processing techniques to analyze the footage and determine the ripeness of the crops as well as count them. The video processing techniques consist of YOLO for object detection, and a deep learning model for fruit ripeness that we will build and train using our unique dataset. Previous work in this field has primarily focused on using ground-based sensors and cameras for crop monitoring, our approach is novel in that it utilizes a drone to capture video footage of the crops from an aerial perspective, allowing for a more comprehensive and efficient method of monitoring crop ripeness and quantity identification.

Keywords: Aerial perspective, Crop management, Crop ripeness, Deep learning, Drone, Heat map generation, Object detection, Real-time monitoring, Video processing, YOLO.

1. Introduction

The importance of crop farming in today's world, and throughout history cannot be overstated, as it can be considered the great initiator of civilization, and to this day provides one of the necessities for life.

Modern machinery, farming techniques, and science have improved the efficiency and yield of farming to the point where it takes a fraction of the effort it used to, but with the ever-increasing demand for affordable, high-quality, easily available food, farmlands become bigger and harder to manage. For that reason, crop management is a critical aspect of modern agriculture, as it can significantly impact crop yield and the sustainability of farming practices.

In recent years, there has been a growing interest in using technology to assist farmers in managing their crops more effectively. One promising approach is the use of drones equipped with deep-learning algorithms to identify the ripeness of crops[1].

In our project, we intend to reduce the amount of time and labor of one of the aspects of crop management, which is crop monitoring for quantity and ripeness, as well as harvesting route optimization. Providing farmers with accurate and timely information about their crops significantly increase their yields and reduce waste.

We plan to achieve the project goals by employing a camera-mounted drone, with a GCU. The drone would scan the current crop status visually and send the recorded data to the GCU, which will be responsible for the drone control, and crop status analysis, using machine learning-assisted image processing algorithms.

To develop our system, we will have to assemble a dataset of images of various crops. This dataset will be used to train our deep-learning model to identify the ripeness of the crops. Finally, we will have to test the system in the field to assess its effectiveness in a real-world setting.

The product we will develop in this project can be easily extended to include other crop management features, e.g., crop sickness identification, for future use.

[1] Bouguettaya, A., Zarzour, H., Kechida, A. *et al.* Deep learning techniques to classify agricultural crops through UAV imagery

1.1. Organization of the project

Section 2 is a summary of the related work our project is based on and other related work in the computer-aided agricultural management field. Section 3 contains the relevant technological, and computer-vision background for our project.

Section 4 elaborates on the project's expected achievements (outcomes, unique features, how we judge success, and more). Section 5 is a detailed explanation of our research process from different aspects (agricultural, ML algorithms, equipment, etc.) including the constraints and challenges we are facing, and our working methodologies. Section 6 is the project's cores containing the product requirements (derived from the research process), system pipeline architecture overview, diagrams, and prototype GUI "screens". Section 7 is where we explain how we are going to evaluate our product and the development verification plan (functional testing). Section 8 contains the references we used while making this project book.

2. Related Work

Knowledge of the distribution of fruits through their detection and location, with different levels of resolution –within a specific tree and at plot level– is of enormous interest in agriculture. Having this information allows harvest and production estimates to be made, which leads to better planning of harvesting, storage, and marketing tasks[2]. With such information, it is also possible to know the spatial distribution of fruits and yield and to relate it to the rest of the variables and factors that influence the management of plantations, such as the strategies of irrigation, fertilization, and pruning, the characteristics, and variability of the soil composition, the topographic characteristics of the plot, the size and structure of the trees, pests, and disease impact, and so on. In addition, knowledge of the georeferenced distribution of fruits along the plot can be a starting point for robotized harvesting, as the harvester robot would have the coordinates of each fruit and could primarily focus on the collection process itself, with a resulting gain in speed and efficiency[3].

Several researchers have considered the issue of crop detection. one of the systems[4]

[2] Bargoti, Suchet, and James P. Underwood. "Image segmentation for fruit detection and yield estimation in apple orchards."

[3] Gené-Mola, Jordi, et al. "Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry".

[4] Q. Wang, S. T. Nuske, M. Bergerman and S. Singh, "Automated crop yield estimation for apple orchards".

developed an apple detection approach to predict the yield. Their detection system was based on the color of the fruit as well as its distinctive specular reflection pattern. Additional information, including the average size of apples, was used to either split regions or remove erroneous detections. A further heuristic employed was to accept as detections only those regions which were considered mostly round. The detection result was then correlated to estimate the final yield. Other[5] systems used the Faster R-CNN architecture for the detection of mangoes, almonds, apples, and a variety of other fruits. Mallikarjuna and Shadaksharaiah[6] were using a novel filter-based model for the classification of tobacco leaves. The filter-based model relies on the estimation of the degree of ripeness of a leaf using a combination of filters and color models. El-Bendary and Nashwa[7] were using an automated multi-class classification approach for tomato ripeness measurement and evaluation via investigating and classifying the different maturity/ripeness stages.

Al-Mashhadani, Zubaidah, and Balasubramaniyan Chandrasekaran[8] OpenCV and HSV color space for detecting tomato ripeness.

3. Background

3.1. Agricultural Background

Agriculture is a vital sector of the global economy, providing food and other essential goods. Crop management is a complex process that requires careful planning, monitoring, and decision-making. One key aspect is monitoring the quantity and ripeness of crops, as well as optimizing harvesting routes, to improve crop yields and reduce waste. Traditionally, farmers have relied on manual methods to do this, but these can be time-consuming and labor-intensive.

3.1.1. Agricultural in Israel

According to data from the Israeli Central Bureau of Statistics, the most common fruit grown in Israel are from the citrus family which includes oranges, grapefruits, and lemons.

[5] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, “Deepfruits: A fruit detection system using deep neural networks”

[6] Mallikarjuna, P. B., D. S. Guru, and C. Shadaksharaiah. "Ripeness Evaluation of Tobacco Leaves for Automatic Harvesting: An Approach Based on Combination of Filters and Color Models."

[7] El-Bendary, Nashwa, et al. "Using machine learning techniques for evaluating tomato ripeness.

[8] Al-Mashhadani, Zubaidah, and Balasubramaniyan Chandrasekaran. "Autonomous Ripeness Detection Using Image Processing for an Agricultural Robotic System."

For this reason, we are going to focus on citrus to match our terrain for easier access to groves.

3.1.2. Citrus Fruit ripeness factors

To judge the ripeness of citrus fruit, the most important factor is color, although texture, smell, taste, and appearance can also be considered. Ripe citrus fruit is typically deep yellow, orange, or red in color, depending on the variety, while unripe citrus fruit may be green or pale in color.

The regular citrus fruit season commences in mid-September (although nowadays fresh lemons and limes may also be available during the summer months) and extends until May the following year.

3.2. Technological/Technical Background

3.2.1. Drone Definition

A drone, also known as an unmanned aerial vehicle (UAV), is a type of aircraft that is operated remotely or autonomously, without a pilot on board. Drones come in a wide range of shapes and sizes and can be used for a variety of purposes, including military and law enforcement operations, search and rescue missions, aerial photography and videography, delivery of goods, and agricultural and environmental monitoring.

Most drones have the following components:

- A central processing unit (CPU) or flight controller, which controls the drone's movements and actions.
- Motors and propellers, which provide lift and allow the drone to move through the air.
- Sensors, such as accelerometers, gyroscopes, and GPS, allow the drone to navigate and maintain stability.
- A communication system, which allows the drone to receive commands and transmit data back to the operator.
- A power source, such as a battery or fuel cell, which provides energy to the drone's systems.

- Cameras and other payloads, which can be mounted on the drone to enable it to perform specific tasks, such as capturing photos or videos or carrying out inspections.

3.2.1.1. DJI Mavic 2 Pro

In our project, we will use DJI Mavic 2 Pro drone (provided by our college).

Figure 1[9]: DJI Mavic 2 Pro

The DJI Mavic 2 Pro's relevant features are summarized in the following table:

Table 1: DJI Mavic 2 Pro spec

<table>
<tr><th></th><th>Feature</th><th colspan="3">Specification</th></tr>
<tr><td rowspan="3">Air Craft</td><td>Weight</td><td colspan="3">2.00lb/907g</td></tr>
<tr><td>Battery / Working time</td><td colspan="3">3850 mAh LiPo 4S / up to 31 minutes</td></tr>
<tr><td>Max Wind Speed Resistance</td><td colspan="3">29-38 kph</td></tr>
<tr><td rowspan="5">Camera</td><td>Camera Sensor</td><td colspan="3">1" CMOS Effective Pixels: 20 million</td></tr>
<tr><td>Camera Lens</td><td colspan="3">FOV (Field of View): about 77°
35 mm Format Equivalent: 28 mm
Aperture: f/2.8–f/11
Shooting Range: 1 m to ∞</td></tr>
<tr><td>Camera ISO Range[10]</td><td colspan="3">Video: 100-6400,
Photo: 100-3200 (auto), 100-12800 (manual)</td></tr>
<tr><td rowspan="2">Camera Video Recording Modes</td><td rowspan="2"><b>Display Standard</b></td><td colspan="2"><b>Details</b></td></tr>
<tr><td><b>Resolution</b> (in pixels)</td><td><b>FPS (Frame per Second)</b></td></tr>
</table>

[9] Picture and Features are taken from the company website: https://www.dji.com/

[10] The International Organization for Standardization range is a measure of the camera's ability to capture light (sensitivity).

		4K	3840x2160	24/25/30
		2.7K	2688x1512	24/25/30/48/50/60
		Full HD	1920x1080	24/25/30/48/50/60/120
	Photo/Video Format	JPEG, DNG(RAW)/MP4, MOV		
	Camera Shutter Speed	Electronic: 8-1/8000s		
Gimbal	Mechanical Range	Tilt: -135–45° Pan: -100–100°		
	Controllable Range	Tilt: -90–30° Pan: -75–75°		
	Stabilization	3-axis (tilt, roll, pan)		
	Max Control Speed (tilt)	120° /s		
	Angular Vibration Range	±0.01°		
Sensing System	System Type	Omnidirectional Obstacle Sensing		
	Forward	Precision Measurement Range: 0.5 - 20 m Detectable Range: 20 - 40 m Effective Sensing Speed: ≤ 14m/s FOV: Horizontal: 40°, Vertical: 70°		
	Backward	Precision Measurement Range: 0.5 - 16 m Detectable Range: 16 - 32 m Effective Sensing Speed: ≤ 12m/s FOV: Horizontal: 60°, Vertical: 77°		
	Upward	Precision Measurement Range: 0.1 - 8 m		
	Downward	Precision Measurement Range: 0.5 - 11 m Detectable Range: 11 - 22 m		
	Sides	Precision Measurement Range: 0.5 - 10 m Effective Sensing Speed: ≤ 8m/s FOV: Horizontal: 80°, Vertical: 65°		

We assume that this type of drone will be suitable to achieve most of our project goals. But if we had other options, we would choose a drone that is more suitable for agricultural work. Such drones need to have a built-in object avoidance system, longer working time, higher transmission distance and be collision-proof (rugged material).

3.2.2. Drone Controller

A drone controller is a device that is used to operate a drone remotely. It typically consists of a transmitter, which is held by the operator, and a receiver that is mounted on the drone. The transmitter sends signals to the receiver, which the drone responds to by performing actions such as moving in a particular direction or altitude or activating certain functions such as taking a photograph or video.

3.2.2.1. DJI Smart Controller

In our project, we will use the DJI Smart Controller (provided by our college) to control and communicate with the DJI Mavic 2 Pro Drone. The communication is RF-based.
The DJI Smart Controller consists of joysticks to adjust the drone angles and speed, and a 5.5-inch built-in screen that streams the drone camera captured video in real time.
There most important feature of this controller, which the default controller does not have, is an HDMI output port. This output port is crucial, without it we won't be able to transmit the captured video to our system for further processing. We will also connect the DJI Smart Controller to our GCU via a USB Type-C or Type-A for communication purposes.

Figure 2: The DJI Smart Controller

The DJI Smart Controller's relevant features are summarized in the following table:

Table 2: The DJI Smart Controller spec

DJI Smart Controller	Specification
Max Transmission Distance (Unobstructed, free of Interference)	**2.400-2.4835 GHz:** 8 km (FCC), 4 km (CE), 4 km (SRRC), 4 km (MIC)
	5.725-5.850 GHz: 8 km (FCC), 2 km (CE), 5 km (SRRC)
Transmitter Power (EIRP)	**2.400-2.4835 GHz:** 25.5 dBm (FCC), 18.5 dBm (CE), 19 dBm (SRRC), 18.5 dBm (MIC)

	5.725-5.850 GHz: 25.5 dBm (FCC), 12.5 dBm (CE), 18.5 dBm (SRRC)
Battery	18650 Li-ion (5000 mAh @ 7.2 V)
Rated Power	15W
Working Time	2.5 hours
Video Output Port	HDMI

3.2.3. External video capture card

An external video capture card is a device that allows you to capture video and audio signals from external sources and transfer them to a computer for storage or editing. It typically connects to the computer through a USB port or a PCI-Express slot, and it may be used to capture video from a variety of sources, including VHS tapes, camcorders, and video game consoles.

3.2.3.1. AV.io HD

AV.io HD is an external video capture card made by Epiphan Video, a company that specializes in professional-grade video capture and streaming solutions. The AV.io HD capture card is designed to capture high-definition video and audio from a variety of external sources, including HDMI, DVI, and VGA. It connects to a computer through a USB 3.0 port and is capable of capturing video at resolutions up to 1080p at 60 frames per second.

Figure 3: AV.io HD Capture card

3.2.4. DJI Windows SDK

DJI Windows SDK is a software development kit (SDK) for Windows that allows developers to create custom applications for DJI drones and other platforms using the DJI SDK. The DJI Windows SDK includes a set of APIs, libraries, and tools that can be used to access and control the flight controls, camera, and other onboard systems of DJI drones and other platforms.

3.2.5. Global Positioning System (GPS)

GPS stands for Global Positioning System. It is a satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The system is operated by the United States government and is freely accessible to anyone with a GPS receiver. It is widely used in navigation systems for cars, boats, airplanes, and smartphones, as well as for tracking the location of assets such as vehicles, ships, and packages.

3.2.5.1. GPS Coordinates[11]

GPS coordinates are used to pinpoint a specific location on the Earth's surface. They are typically presented in the form of latitude and longitude and are often expressed in degrees, minutes, and seconds (DMS).

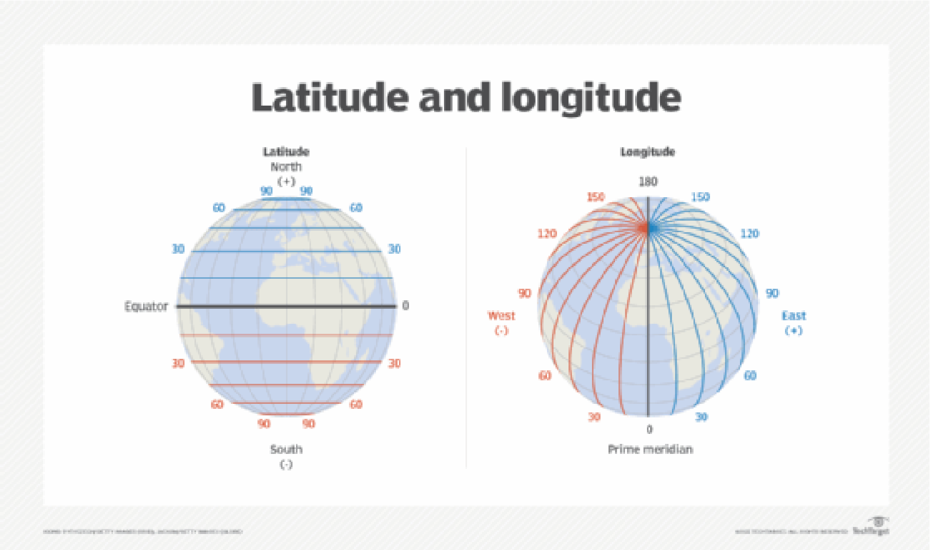


Figure 4: Latitude coordinates measure distance north and south from the equator. Longitude coordinates measure distance east and west from the prime meridian.

The format for GPS coordinates typically looks like:

$$Latitude: xx^{\circ}\ xx'\ xx''\ N\ (or\ S)$$

$$Longitude: yy^{\circ}\ yy'\ yy''\ E\ (or\ W)$$

[11] https://www.techtarget.com/whatis/definition/GPS-coordinates

It is important to note that the "N" or "S" after the latitude indicates whether the location is in the northern or southern hemisphere, while the "E" or "W" after the longitude indicates whether the location is in the eastern or western hemisphere.

The DJI API, expresses the GPS coordinates in a decimal format.

To convert GPS coordinates from degrees, minutes, and seconds (DMS) to decimal format, we use the following formulas:

$$Decimal\ Latitude\ =\ Degrees\ +\ \left(\frac{Minutes}{60}\right)+\ \left(\frac{Seconds}{3600}\right)$$

$$Decimal\ Longitude\ =\ Degrees\ +\ \left(\frac{Minutes}{60}\right)+\ \left(\frac{Seconds}{3600}\right)$$

Note that for the longitude, if the coordinates are West we need to put minus sign before the calculation and for the latitude, if the coordinates are South we need to put the minus sign before the calculation.

For example, the coordinates for New York City's Central Park in DMS format are:

$Latitude$: $40°\ 46'\ 51.9"\ N\ (40\ degrees, 46\ minutes, 51.9\ seconds\ North)$

$Longitude$: $73°\ 58'\ 27.2"\ W\ (73\ degrees, 58\ minutes, 27.2\ seconds\ West)$

In decimal format, the same coordinates would be:

$$Decimal\ Latitude\ =\ 40\ +\ \left(\frac{46}{60}\right)+\ \left(\frac{51.9}{3600}\right)\ =\ 40.78054$$

$$Decimal\ Longitude\ =\ 73\ +\ \left(\frac{58}{60}\right)+\ \left(\frac{27.2}{3600}\right)\ =\ -73.9743$$

3.2.6. System Connectivity Architecture

In the last few sections, we explained how each component of our system is working “stand-alone”. To explain the whole system connectivity architecture, we need to analyze our system connectivity requirements.

Our system has two basic communication requirements:

1) The GCU should be able to receive flying telemetry and the video captured by the drone in real time.

2) Automation - The GCU should be able to send commands to the drone to which the drone needs to react in real time.

Both requirements can be fulfilled by connecting the DJI Smart Controller to the drone (via RF) and to the GCU (via USB cable). In the GCU we are going to run the software which will use the DJI Windows SDK API to get the telemetry and send commands to the drone. Also, In this method, the video output will be of a low quality so it will be better to use our AV.io external video capture card to capture the video from the smart controller and transfer it to the GCU.

In figure 5, we can see the system connectivity architecture:

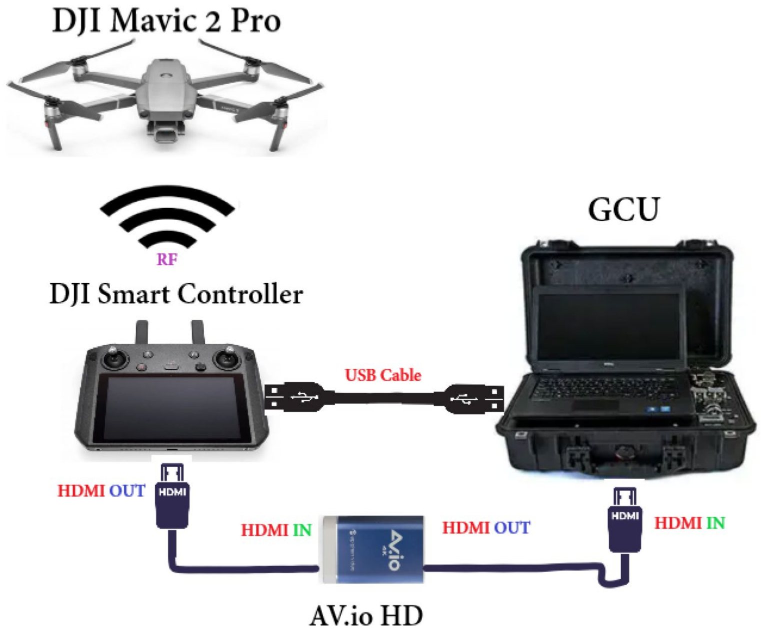


Figure 5: Our system connectivity architecture diagram

3.3. Computer Vision Background

3.3.1. Computer Vision (CV)

Computer vision is a field of artificial intelligence that focuses on enabling computers to interpret and analyze visual data from the world around them.

Computer vision systems use algorithms and machine learning techniques to analyze images and videos and extract information from them. This can be used for a wide range of applications, including self-driving cars, security systems, and medical diagnosis.

3.3.2. Optical Flow

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Optical flow can also be thought of as the distribution of the apparent velocities of objects in an image. It is a 2D vector field where each vector is a displacement vector that describes the movement of a point from one position in the image to another position in the same image. In other words, optical flow is a way to represent the motion of pixels between successive frames in a video. It is used in many computer-vision tasks, such as object tracking, image alignment, and structure from motion.

3.3.2.1. The Optical Flow of DJI Mavic 2 Pro

The DJI Mavic 2 Pro uses an optical flow sensor to measure the motion of pixels in the images captured by its bottom-facing camera. This sensor is used to provide additional stability and accuracy when flying the drone, particularly when flying at low altitudes or in environments with limited GPS coverage.

The optical flow sensor is also used in combination with other sensors, such as the drone's ultrasonic sensors and inertial measurement unit (IMU), to provide a more complete picture of the drone's surroundings and movement. This allows the drone to maintain a stable hover and navigate accurately in complex environments.

3.3.3. Machine Learning (ML)

Machine learning is a type of artificial intelligence that involves training algorithms on data sets so that they can learn to recognize patterns and make decisions based on those patterns. Machine learning algorithms can be trained on a variety of tasks, including classification, regression, and clustering.

3.3.4. Deep Learning (DL)

Deep learning is a specific type of machine learning that involves the use of artificial neural networks with multiple layers.

3.3.4.1. Neural Network (NN)

A neural network is a type of machine-learning algorithm that is inspired by the way the human brain works. It is made up of many interconnected "neurons," which are inspired by the cells in the brain that transmit information.

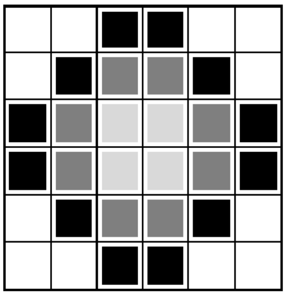
Neural networks can be used for a wide range of tasks, including image and speech recognition, natural language processing, and decision-making. They are particularly well-suited for tasks that involve complex patterns and relationships in data, such as image and speech recognition.

3.3.4.2. Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a type of feedforward neural network made up of multiple layers specifically designed for processing data that has a grid-like topology, such as a digital image.

3.3.5. Digital Image

A digital image is a representation of a visual scene or object in the form of a two-dimensional array of pixels. Each pixel in the image represents a specific color, and the combination of all the pixels in the image creates the overall visual appearance of the image.



```
11 11 00 00 11 11
11 00 01 01 00 11
00 01 10 10 01 00
00 01 10 10 01 00
11 00 01 01 00 11
11 11 00 00 11 11
```

Figure 6: Digital Image example [(11, “white”), (01, “grey”), (00, “black”)]

3.3.6. Object Detection

Object detection is one of the classical problems in computer vision where you work to recognize what and where — specifically what objects are inside a given image and where they are in the image. The problem of object detection is more complex than classification, which also can recognize objects but does not indicate where the object is in the image. In addition, classification models do not work well on images with more than one object.

3.3.6.1. Faster R-CNN[12] Algorithm

Faster R-CNN is an extension of the R-CNN (Regions with CNN features) object detection algorithm and is designed to be faster and more accurate than R-CNN.

The Faster R-CNN is composed of two modules. The first module is a deep fully convolutional network that proposes regions (RPN), and the second module is the Fast R-CNN detector that uses the proposed regions. The entire system is a single, unified network for object detection (Figure 7).

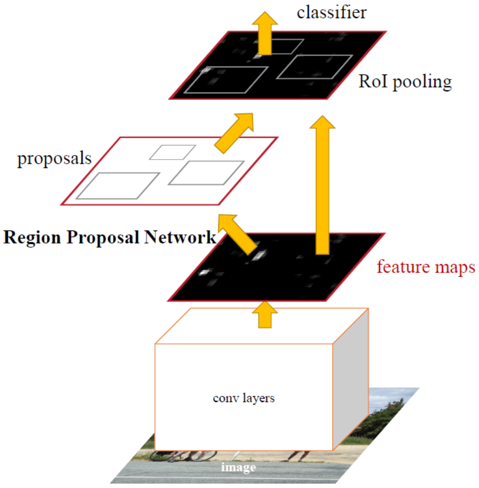


Figure 7: Faster R-CNN Architecture. The RPN module serves as the 'attention' of this unified network.

3.3.6.2. You Only Look Once (YOLO[13]) Algorithm

YOLO (You Only Look Once) is a real-time object detection algorithm that is designed to be fast and accurate.

YOLO “only looks once” at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes.

[12] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks."
[13] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection."

With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance.

YOLO has several benefits over other object detection methods:

- YOLO sees the entire image during training and test time, so it implicitly encodes contextual information about classes as well as their appearance.
- YOLO learns generalizable representations of objects.
  In other words, if YOLO is trained on natural images (such as photographs) and then tested on artworks (such as paintings or drawings), it can recognize the objects in the artwork even though they may be represented differently than in the natural images. This is because YOLO has learned a generalizable representation of the objects, rather than a representation that is specific to a particular context.
- YOLO is fast and suitable for real-time applications.

In figure 8, we can see an example of the YOLO detection system. The system (1) resizes the input image to $448x448$, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model’s confidence.

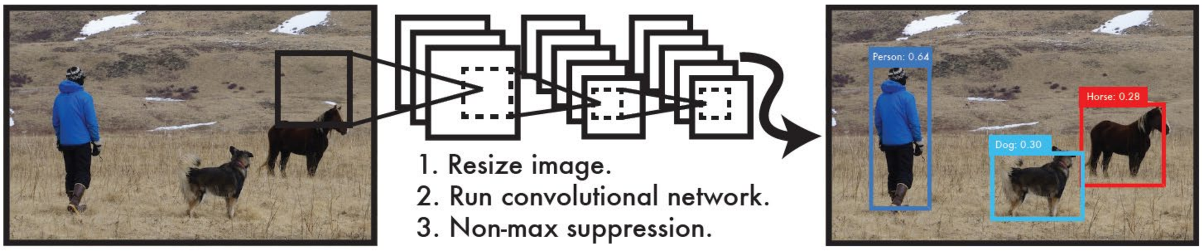


Figure 8: The YOLO Detection System

3.3.6.3. Faster R-CNN vs YOLO

YOLO shares some similarities with R-CNN. Each grid cell proposes potential bounding boxes and scores those boxes using convolutional features.
However, YOLO puts spatial constraints on the grid cell proposals which helps mitigate multiple detections of the same object and proposes far fewer bounding boxes.

Another aspect is that Faster R-CNN focuses on speeding up the R-CNN framework by sharing computation and using neural networks to propose regions instead of Selective Search. While Faster R-CNN offers speed and accuracy improvements over R-CNN, it is still not capable of a real-time performance like YOLO.

4. Expected Achievements

4.1. Outcomes

The main goal of this project is to develop a system for crop management in the aspect of monitoring the crop quantity and ripeness and optimizing its harvesting routes. We believe that this will reduce the time and labor needed to manage crops, which will result in a significant increase in crop yields and waste reduction.

We expect that our system will be able to scan a given farmland and generate a detailed analysis of its crop quantity and ripeness in a sort of "heat-map" output. This can be achieved by getting the farmland map, crop type, and ripeness threshold from the farmer (UI) as input. Our system will generate the drone flying route (by analyzing the map) and start scanning the farmland with the drone. While the drone is flying, the GCU communicates with it for progress evaluation, and route adjustments that were done by analyzing the drone video stream in real-time with our novel video processing algorithm pipeline. The pipeline starts with YOLO for object detection, tracking, and segmentation purposes. YOLO's output, segmented objects, are the input for our CNN that will classify the ripeness of each object (the model will be trained with a dataset that we will create). Finally, we will generate a 2D "heat map" that indicates where clusters of ripe crops are, the most efficient route to harvest them, and a report of the scan (which will be available at the UI).

4.2. Unique Features

4.2.1. Drone

4.2.1.1. Communication

One of the unique features of our project is to establish and maintain a stable communication link between the drone and the Ground Control Unit (GCU), to ensure that the video stream and other data captured by the drone are transmitted and analyzed in real-time. This requires addressing issues such as distance and range, interference, and latency in the communication link, to minimize data loss and ensure a smooth and efficient operation of the system.

4.2.1.2. Programming

Another unique feature, related to drone, is programming a DJI drone using the DJI Windows SDK. it can pose several challenges, such as complexity in understanding the capabilities and limitations of the drone and SDK, ensuring the stability and safety of the drone during flight, and dealing with limited documentation and support. Additionally, compatibility issues between the SDK and the operating system or programming language being used may also arise.

4.2.2. Citrus Fruits Aerial Dataset

Collecting data for a citrus fruits aerial dataset can be challenging as it requires access to large orchards or farms, obtaining permission to fly drones or aircraft for aerial photography, and ensuring that the images captured contain a sufficient number and variety of citrus fruits for training a deep learning model. Additionally, it may be difficult to ensure consistent lighting and weather conditions for the images, which can affect the quality of the dataset. Annotating the images to label the different types of citrus fruits and their locations within the images can be time-consuming and require specialized knowledge of citrus fruits. Ensuring diversity in the dataset, such as different types of citrus fruits, different stages of ripeness, different backgrounds, different lighting conditions, and different weather conditions, can be difficult to achieve and may require multiple data collection campaigns across several seasons, this help to increase the robustness of the model and to be able to generalize well. Ensuring that the images are of high quality and that the labels are accurate and consistent is crucial for training a deep learning model. It may be necessary to perform quality control checks on the images and labels to ensure their quality and consistency.

4.3. Criteria for Success

- We managed to build a sufficient Dataset that contains enough images of citrus fruits in different positions, ripeness levels, and light levels. We will judge that based on the model's ability to learn (not getting to over-fitting in the beginning).
- Our citrus ripeness classification CNN model has at least 80% accuracy.
- A successful POC of the product on an actual citrus orchard.

- The product video processing pipeline can be easily extended with more features (e.g., crop illness detection) and more crop types – we can judge that based on our experience to integrate the ripeness model into the rest of the pipeline.

5. Research/Engineering Process

Our project research process can be divided into three parts: agricultural research, technical research, and algorithmic research.

5.1. Agricultural Research

To design an efficient agricultural system we concluded that we first need to deepen our understanding of the field of agriculture, regarding topics such as:

- What are the methods used for estimating crop ripeness?
- What tooling is required to perform accurate estimations in a timely fashion?
- Which crops grow in Israel (for dataset and testing purposes)?
- When do the various crops ripen?
- How are crop field layouts designed?

To answer these questions, we consulted many various sources, such as agricultural magazine articles, research papers, and educational videos.
We met multiple times a week, and each meeting jointly tackled one of the abovementioned questions, recapping the research so far at first, then splitting it into individual research, followed by a short discussion about the findings, and planning for future meetings.

5.1.1. Constraints and Challenges

While conducting the research we came to be aware of multiple constricting factors, such as short timeframes for dataset gathering, closely packed crop layouts, with narrow flight paths for scanning, and the complexity of designing a flight path for a typical crop layout.
In addition, we found out that not all types of crops can be accurately assessed using visual

means only.

5.1.2. Decisions

The conclusion we came to following our agricultural research was that the most fitting crop for our research was citrus fruits, which are abundantly present in Israel, have relatively long timeframes for data gathering, and are usually planted in long rows.
In addition, the ripeness of citrus fruits can be assessed accurately using a camera.

5.2. Technological Research

The research process for the technological side of our project was fairly similar to the agricultural side, however, having more experience in the field as SE students, we could concentrate more on the feasibility of using and ease of use of each technological aspect, having a good general idea about which technologies we would like to use in our project as early as the initial idea planning stages.
That being said we still had to deepen our understanding of how drones operate and communicate, and which frameworks are available for developing complex scanning routines. In addition, we investigated what kind of hardware would be required for training a neural network, sending commands to the drone, and receiving a video stream from the drone.

5.2.1. Constraints and Challenges

From our technological research, several constraints became apparent, such as drone flight time, which is usually limited only to several minutes, and up to half an hour at best on higher-end models, as well as a relatively short reception distance for commands, and a maximum windspeed over which the drone cannot fly in a stable manner.
We also found out that even though streaming the video from the drone controller to the server over a USB cable is possible, the quality of the video stream might be severely degraded.

5.2.2. Decisions

The conclusions drawn from the technological research pointed at the possibility of using a different model drone during the implementation part of the project, as well as using a dedicated video capture card for video instead of a simple USB cable.

For model training, we decided that the hardware offered by Google for free under the Collab platform would be sufficient at this point in the research, and if more computational power is required, it is readily available on the same platform for a reasonable price.

5.3. Algorithmic Research

For algorithmic research, we started by planning out a rough pipeline for the algorithms and models used and started looking for papers and implementations that best suit our needs, and are compatible with each other.

We looked specifically for models which can operate in real-time, and process video taken from a moving viewpoint, other factors such as outputting segmented results, rather than bounding boxes were considered, especially when the required computation time isn't significantly higher.

Additionally, if we were to develop a machine learning model, we would need a fitting dataset for the job, which led us to search through multiple publicly available dataset sources.

5.3.1. Constraints and Challenges

We would like our product to be able to detect if certain portions of a scan were faulty, and schedule a rescan within the same routine, therefore, at the very least the fruit detection model in the detection and assessment pipeline will have to run in real-time, which is a very resource intensive task.

In addition, after scouring the currently publicly available dataset databases, we found out that a fitting dataset of citrus fruit, where the fruit was photographed throughout its growth and ripening lifecycle, is not available.

5.3.2. Decisions

After considering the currently available detection models, we decided to pick YOLO, as opposed to other detection algorithms such as Mask R-CNN as it provides accurate results in real-time, with the bonus of segmented images as output, ready for further analysis by a ripeness assessment model that we will be trained on a new dataset which fits our needs (we will build the dataset alone on the second part of the project).

5.4. Methodology and Development Process

The methodology we chose for our development process is Agile.
We think that it fits the timeframe, scale, and required flexibility of our project well, as well as offering a clear structure for planning out the development steps.

The development process will consist of the following steps:

- Taking pictures of fruit for a ripeness assessment model dataset.
- Developing a basic drone communication and control module.
- Developing a module for planning out a scanning route given a map.
- Training a YOLO model for fruit detection and segmentation.
- Developing a CNN-based model for fruit ripeness assessment given segmented image.
- Integrating full scan and detection pipeline.
- Creating a simple UI for defining the paths clear for flying on a map.
- Creating a simple UI for scheduling scans and displaying scan results.

The development process will consist of developing and integrating every sub-module within each cycle, allowing for changes to be made if required.
We decided to start with collecting the pictures required for the dataset because it is time critical, as fruit ripeness cycles are seasonal.

6. Product

6.1. Requirements

6.1.1. Functional

#	Requirement
1	The system would be able to receive a farmland map from the user.
2	The system would be able to receive a crop type from the user.
3	The system would support two different crop types for the POC.
4	The system would be able to receive a ripeness threshold from the user.
5	The system would be able to plan a scanning route of the farmland from the map the user provided.
6	The system would be able to receive flying telemetry from the drone in real-time.
7	The system would be able to control the drone in real-time (position, speed, etc).
8	The system would be able to identify crop objects on the video received from the drone in real-time.
9	The system would be able to classify the crop object ripeness.
10	The system would be able to identify errors in the scanning route.
11	The system would be able to adjust the drone scanning route.
12	The system would be able to store the scan data.
13	The system would be able to generate crop status report from the scan data.

6.1.2. Non-Functional

#	Requirement
1	The system UI would be easy to use.
2	The farmland map of the user would be a set of GPS coordinates.
3	The ripeness threshold indicates the % of ripeness (1-100) the user desire.
4	The system crop types are orange and lemon.
5	The crop status report will contain a “heat map” of ripened crops in the farmland.
6	The crop status report will contain the shortest harvesting route for the ripened fruits.
7	The crop status report will contain a report with the total crop quantity and ripe crop quantity.

6.2. Architecture Overview

The Architecture of our project consists of the following components:

- Drone which flies throughout the field, scanning the crops.
- Ground Control Unit (GCU), which controls the drone, and processes the video received from the drone.
- Graphical User Interface (GUI) for defining the flyable areas, scheduling scans, and displaying scan results.
- Machine learning pipeline for processing the video, consisting of YOLO and our model.

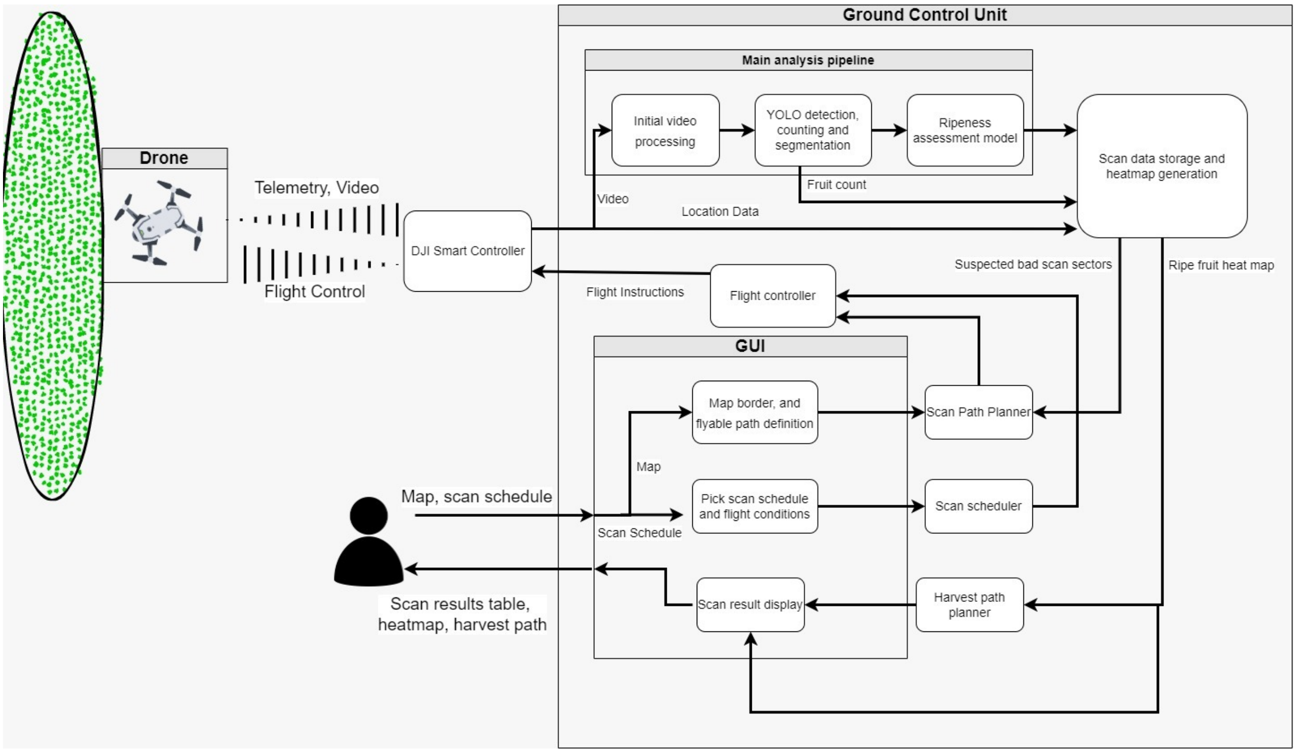


Figure 9: Our system architecture

6.2.1. User Interface (UI)

The UI will allow three main functions:

1) Provide a GPS map of the farmland, and mark on the map scan relevant features, such as farm borders, flyable areas, and crop areas for scanning. This information will be further used for planning out a scan path for the drone by using a traversal algorithm and building a crop heatmap.

2) Scheduling scans, whether manual or periodic, with the option to add triggers and conditions such as postponing a scan due to bad weather conditions for a flight.

3) Displaying of scan results, the results will consist of a table with information such as total fruit count, ripe fruit count, and percentage and areas that required additional scans. In addition, after a complete scan we will generate and display a heat map of ripe fruit sectors overlayed on top of the map provided by the user, with a suggested optimal path for the harvest of hotspots.

6.2.2. Ground Control Unit (GCU)

The GCU will be an on-location server with a wired connection to the DJI Smart Controller, which will provide the following functions:

- A local instance of the management UI for easy control over the system.
- Calculation of the flight path based on the provided map.
- Direct control of the drone through the DJI Smart Controller API.
- Reception and processing of the video and telemetry sent from the drone.
- Real-time scan error detection and adjustment of scan path for rescan.
- Running the ML-based ripeness assessment pipeline on the received video.
- Calculation of total fruit amount and ripe fruit percentage.
- Generation of ripe fruit heat map.
- Output of the results into the UI.
- Counting of ripe fruits per scan sector.
- Generation of optimal ripe fruit harvesting path.

The GCU is the “brain” of the operation, handling the drone control, image processing, and UI aspects of our project. The GCU will also tie the ripe fruit detections to the current drone location on the planned flight path, thus allowing us to estimate ripe fruit concentrations for calculating an optimal harvesting path.

6.2.3. Models Architecture/Pipeline

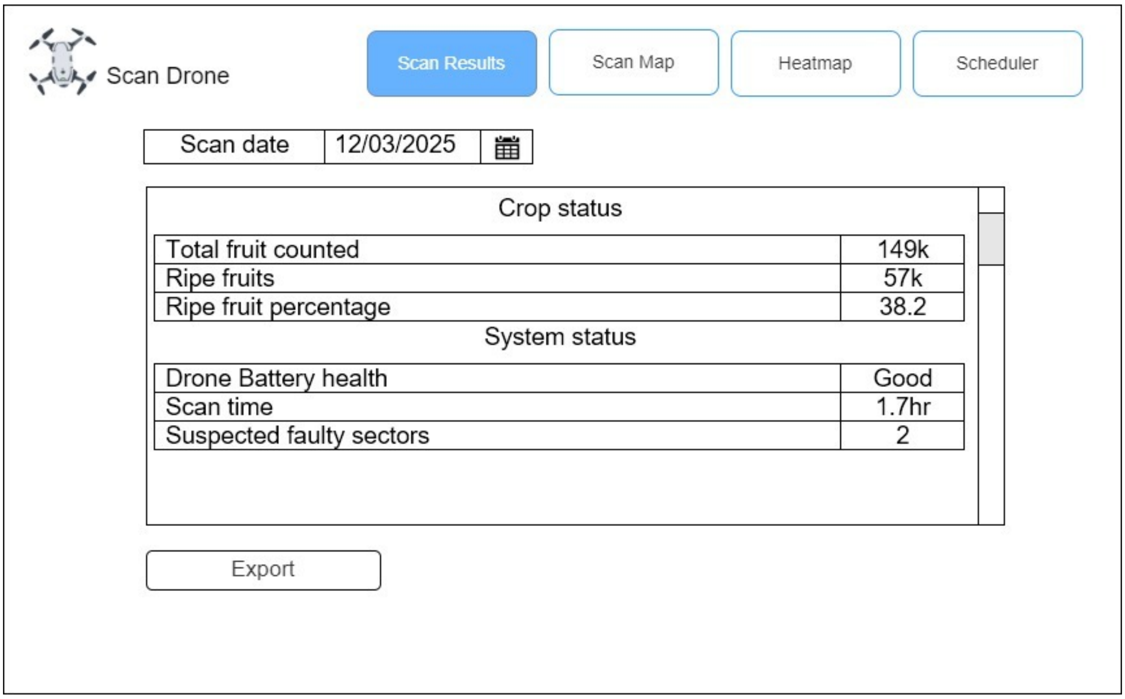
The model pipeline will consist of a video stream input from the drone, which will be processed by a YOLO model which was trained on fruit detection for the detection, counting, and segmentation of the individual fruits captured on video. The individual YOLO detections will in addition serve as the counting method for fruits of any ripeness level. After detection and segmentation by YOLO, the segmented images are passed into our CNN-based model for fruit ripeness assessment from segmented fruit images, which will be trained on a dataset consisting of fruit in different ripeness stages.

The dataset for training the ripeness model will be built by us during the second stage of the project, by photographing fruit throughout different ripeness stages and segmenting and tagging the photographs (we will probably also use data augmentation too).

6.2.4. **Drone**

The drone will oversee the physical aspect of our project, by flying through the fields on the previously defined flight paths. The drone and the GCU will be in constant communication through the DJI Smart Controller, the GCU "telling" the drone where to fly, based on the previously calculated flight path, and possible path adjustment due to faulty scans, and the drone in turn providing the GCU with the scan video, as well as telemetry data such as location, velocity, and battery status. The communication between the drone and the GCU, as well as the sending of control commands will be achieved by implementing the API provided by DJI. We think that a flexible and easy-to-use drone control and communication API is key for project success, therefore, in the second part of the project we might opt for changing the exact drone model if it better suits our needs.

6.3. Graphical User Interface (GUI) Prototype



Scan date 12/03/2025

Crop status

Total fruit counted	149k
Ripe fruits	57k
Ripe fruit percentage	38.2

System status

Drone Battery health	Good
Scan time	1.7hr
Suspected faulty sectors	2

Figure 10: Scan Results

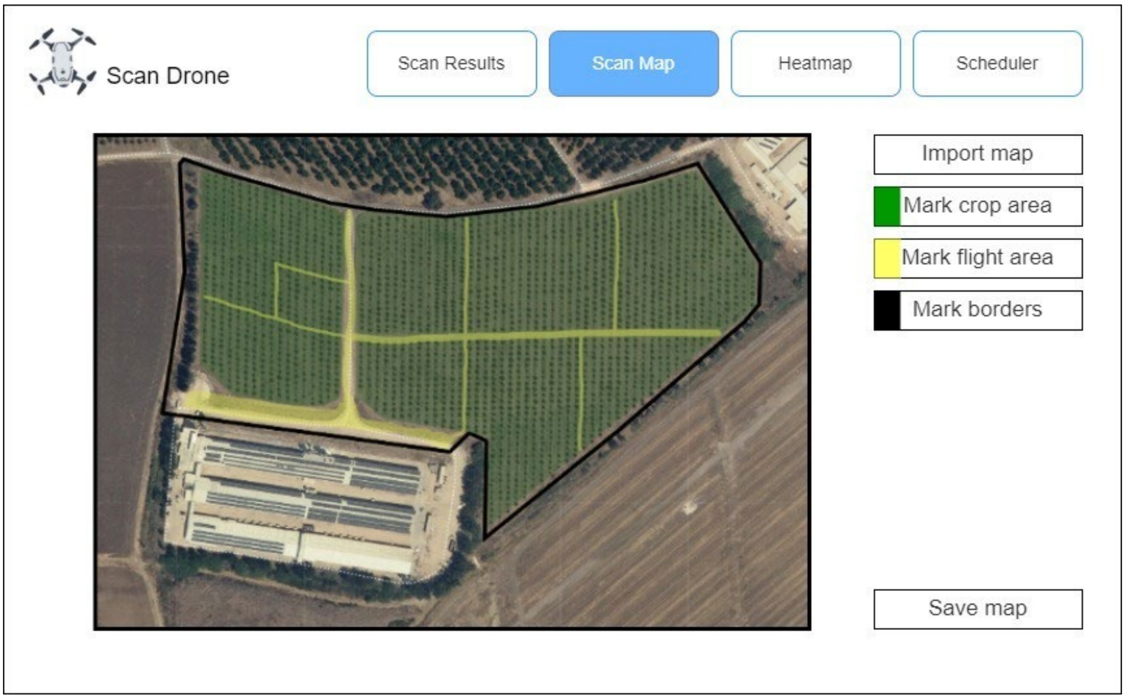


Figure 11: Scan Map

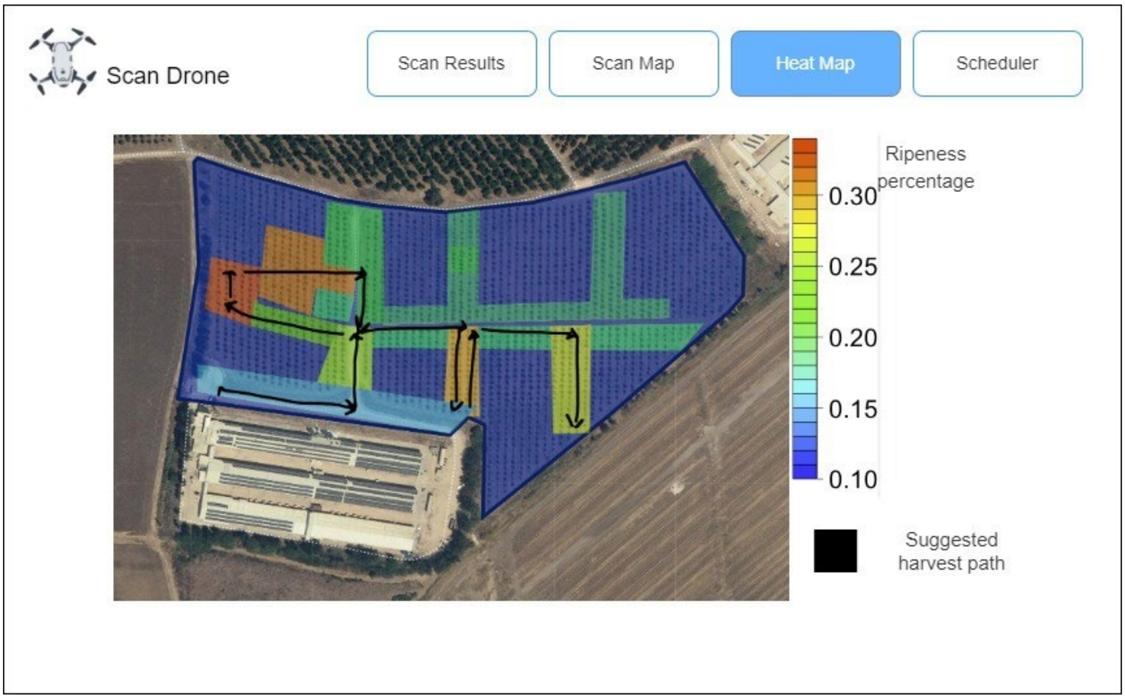
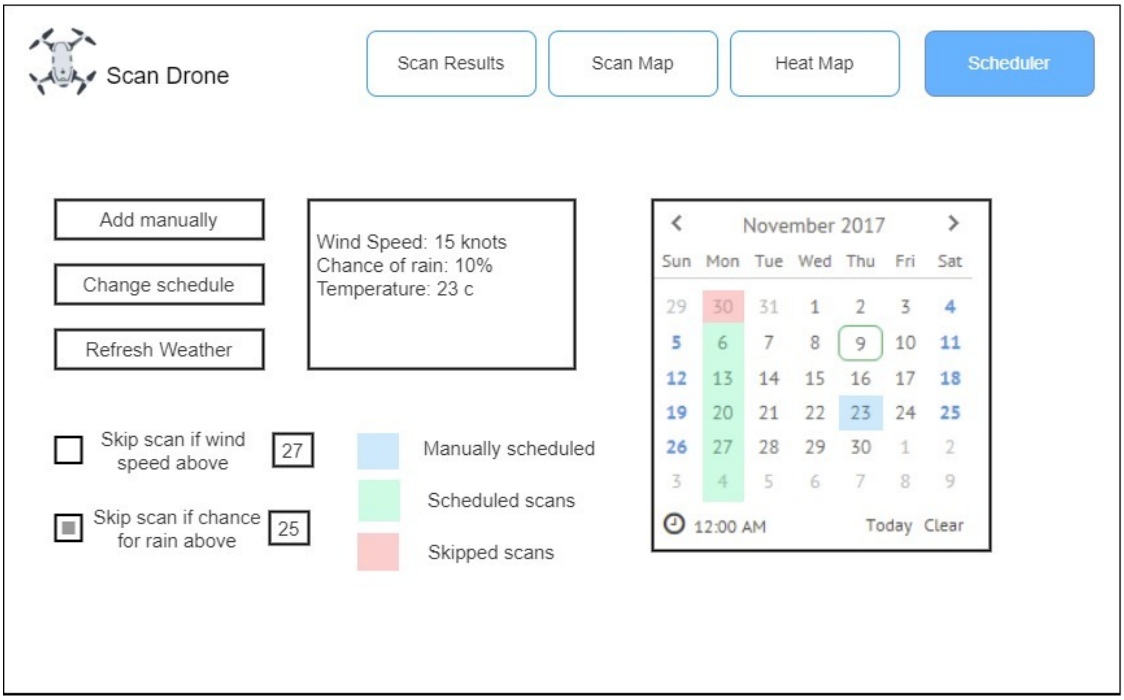


Figure 12: Heat Map



Wind Speed: 15 knots
Chance of rain: 10%
Temperature: 23 c

□ Skip scan if wind speed above 27

☑ Skip scan if chance for rain above 25

🟦 Manually scheduled

🟩 Scheduled scans

🟥 Skipped scans

November 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

12:00 AM

Figure 13: Scheduler

7. Evaluation and Verification

7.1. Evaluation Plan

We will evaluate our product from two different aspects. The first aspect is that our CNN ripeness classification model can correctly classify the ripeness of the chosen crops. It is the most important aspect because we are building this model from scratch as well as the dataset this model is trained with. The second aspect is that we have a successful POC of the whole system scanning a farmland in real-time. The farmland will be analyzed manually by us, and we will compare it to the system output. If we will see in the comparison that our system has a high success rate (above 80% right) this will be the indication for a successful POC.

7.2. Verification Plan

Modules: UI, Communication (Drone & GCU), Path Planner (beginning with a certain route, and adusjt in rt), Video Processing pipline (yolo standalone, cnn standalone), Output generation testing.

Test	Tested Module	Tested Function	Expected Result
1	**User Interface**	**Map loading**	**Map loaded successfully**
2	**User Interface**	**Map marking**	**Marking and area will return GPS coordinates and area type**
3	**User Interface**	**Flight scheduling**	**UI scheduled flights trigger scans**
4	**User Interface**	**Displaying of data**	**UI will be able to pull and display scan results**
5	**Ground Control Unit**	**Drone control**	**Drone will respond to commands sent from the GCU**

6	Ground Control Unit	Connection to drone	Connection is stable and data is exchanged reliably with minimal delay
7	Ground Control Unit	Reception of video from drone	GCU receives live feed from drone, and passes it to the model pipeline
8	Ground Control Unit	Path planning	GCU able to calculate scan and harvest paths based on the map and the scan results respectively
9	Ground Control Unit	Running the model pipeline	GCU hardware runs the predictions at 1 prediction per second at least, doesn’t crash or get stuck
10	Ground Control Unit	Detection of faulty path sections	If no fruit are detected for a 100m sector of the path, it is marked as faulty and rescanned
11	Ground control Unit	Generation of heatmap	Heatmap generated from fruit count and localization data
12	Model pipeline	YOLO detection and segmentation	Detects and segments the fruit

			from the video stream with 80%+ accuracy
13	**Model pipeline**	**CNN ripeness model predictions**	**Ripeness predictions on segmented images from YOLO have an 80%+ accuracy**
14	**Drone**	**Localization**	**The drone reports accurate location in relation to GCU during flight**
15	**Drone**	**Video feed**	**Video sent over reliably and in high quality for detection**
16	**Drone**	**Reception**	**Drone operation range large enough to cover a crop field**
17	**Drone**	**API**	**Drone has an API allowing for programmatic control**

8. References

1. Bouguettaya, A., Zarzour, H., Kechida, A. *et al.* Deep learning techniques to classify agricultural crops through UAV imagery: a review. *Neural Comput & Applic* **34**, 9511–9536 (2022).

2. Bargoti, Suchet, and James P. Underwood. "Image segmentation for fruit detection and yield estimation in apple orchards." *Journal of Field Robotics* 34.6 (2017): 1039-1060.

3. Gené-Mola, Jordi, et al. "Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry." *Computers and Electronics in Agriculture* 169 (2020): 105165.

4. Q. Wang, S. T. Nuske, M. Bergerman and S. Singh, "Automated crop yield estimation for apple orchards", *Proc. 13th Int. Symp. Exp. Robot.*, pp. 745-758, 2012.

5. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," Sensors, vol. 16, no. 8, p. 1222, 2016.

6. Mallikarjuna, P. B., D. S. Guru, and C. Shadaksharaiah. "Ripeness Evaluation of Tobacco Leaves for Automatic Harvesting: An Approach Based on Combination of Filters and Color Models." Data Science. Springer, Singapore, 2021. 197-213.

7. El-Bendary, Nashwa, et al. "Using machine learning techniques for evaluating tomato ripeness." Expert Systems with Applications 42.4 (2015): 1892-1905.

8. Al-Mashhadani, Zubaidah, and Balasubramaniyan Chandrasekaran. "Autonomous Ripeness Detection Using Image Processing for an Agricultural Robotic System." 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2020.

9. https://www.dji.com/

10. https://photographylife.com/what-is-iso-in-photography

11. https://www.techtarget.com/whatis/definition/GPS-coordinates

12. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).

13. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.