

# Agri Robot

by

**Reem Alkhatib 202202927**

**Software Engineering**

*Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor in Computer Engineering*

### Supervised By: Dr. Iman Haidar

Spring 2024/2025

## Table of Contents

[Preface 1](#_Toc194839094)

[Table of Contents 4](#_Toc194839096)

[List of Figures 5](#_Toc194839097)

[List of Tables 7](#_Toc194839098)

[I- Introduction 8](#_Toc194839099)

[1-Overview 8](#_Toc194839100)

[2-Background 8](#_Toc194839101)

[2.2- Objectives: 10](#_Toc194839102)

[2.3- Motivation : 11](#_Toc194839103)

[2.4- Applications: 12](#_Toc194839104)

[II-Literature Review 14](#_Toc194839105)

[1-Introduction: 14](#_Toc194839106)

[2-Existing Work 14](#_Toc194839107)

[3-Alternative Design : 23](#_Toc194839109)

[III- Project Planning: 46](#_Toc194839111)

[Feasibility Study 47](#_Toc194839112)

[Constraints: 48](#_Toc194839113)

[Standards: 49](#_Toc194839114)

[Team Members Tasks 50](#_Toc194839115)

[Software Model : 51](#_Toc194839116)

[Project Issues : 52](#_Toc194839117)

[Tools /Technology: 53](#_Toc194839120)

[Milestones: 54](#_Toc194839121)

[Requirements 57](#_Toc194839123)

[III- Design: 58](#_Toc194839111)

[Bibliography: 63](#_Toc194839125)

## List of Figures

[Figure 1 Burnt Areas in South Lebanon(October 2024 assessment ) -Data Source](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338946)

[:European Forest Fire Information System 11](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338946)

[Figure 2 Faster R-CNN Architecture [17] 23](#_bookmark15)

[Figure 3 Single Shot Mutibox Detection Architecture[19] 25](#_bookmark17)

[Figure 4 Mask R-CNN Architecture[22] 27](#_bookmark20)

[Figure 5 RentinaNet Architecture[24] 29](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338950)

[Figure 6 Raspberry Pi 4 Model B 39](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338951)

[Figure 7 ESP32 microcontroller 40](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338952)

[Figure 8 ESP8266 microcontroller 41](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338953)

[Figure 9 Raspberry Pi camera 43](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338954)

[Figure 10 ESP32 Camera 44](#_bookmark27)

[Figure 11 USB camera 44](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338956)

## List of Tables

[Table 1: state of the art research comparison 20](#_bookmark13)

[Table 2 Yolov8 and Faster RCNN Comparative Analysis[18] 24](#_bookmark16)

[Table 3 YOLOv8 versus SSD comparative analysis[20] 26](#_bookmark18)

[Table 4 Performance comparison between YOLOv8 and SSD[21] 26](#_bookmark19)

[Table 5 Comparative Analysis between Mask R-CNN and YOLOv8[23] 28](#_bookmark21)

[Table 6 Frontend Comparison Table [29] 31](#_bookmark22)

[Table 7 Backend Comparison Table [30] 32](#_bookmark23)

[Table 8 Communication Comparison Table [31] 34](#_bookmark24)

[Table 9 Database Comparison Table[32] 35](#_bookmark25)

[Table 10 Cloud Platforms Comparison Table[33] 37](#_bookmark26)

[Table 11 Cameras Comparison Table 45](#_bookmark28)

[Table 12 Feasibility Study 69](#_bookmark46)

[Figure 13 YOLO Architecture[19] 49](#_bookmark34)

[Figure 14 YOLO models comparative analysis[26] 50](#_bookmark35)

[Figure 15 YOLOv8 architecture, divided into four parts: Backbone, Neck, Head and](#_bookmark36) [Loss[27] 51](#_bookmark36)

[Figure 16 Dataset classes 53](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338961)

[Figure 17 Raw image 54](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338962)

[Figure 18 Annotated raw image in YOLOv8 Format 54](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338963)

[Figure 19 Data Preprocessing leaf Disease Dataset 55](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338964)

[Figure 20 classes for tomato ripeness Dataset 56](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338965)

[Figure 21 Raw image for tomato dataset 57](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338966)

[Figure 22 Annotated raw image for tomato dataset in YOLOv8 Format 57](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338967)

[Figure 23 Data Preprocessing tomato Dataset 58](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338968)

[Figure 24 SPEC 7 in 1 Sensor 60](#_bookmark41)

[Figure 25 HC-SR04 sensor 61](#_bookmark42)

[Figure 26 SG90 Micro Servo 62](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338971)

[Figure 27 Raspberry Pi Model B 63](#_bookmark43)

[Figure 28 Raspberry pi camera 64](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338973)

[Figure 29 Arduino Mega Board 65](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338974)

[Figure 30 SRD-5VDC-SL-C Relay Module 66](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338975)

[Figure 31 DC Motor 66](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338976)

[Figure 32 Monster Moto Shield Module 67](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338977)

[Figure 33 Rover's Wheeled Chassis 67](#_bookmark44)

[Figure 34 RS-485 Module 68](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338979)

[Figure 35 ER Diagram 73](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338980)

[Figure 36 Users Accessibility Flowchart 78](#_bookmark55)

[Figure 37 Rover and Website Relationship Flow Chart 79](#_bookmark56)

[Figure 38 Home page 80](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338983)

[Figure 39 Login Page 81](file://localhost/C:/Users/user/Desktop/Final%20Year%20Project%20Final%20Version.docx%23_Toc186338984)

[Figure 40 Sign Up Page 81](#_bookmark58)

[Figure 41 About Us Page 82](#_bookmark59)

## Introduction

### Overview

Agriculture is essential for any society, playing a vital role in the growth of a country's economy and its development. While some countries are utilizing latest agricultural approaches, majority still rely on traditional methods and techniques that are labor intensive and time consuming. The use of traditional approaches is failing to serve the high demand we are facing nowadays due to population growth. These problems highlight the urgent need for the use of new technologies to address these challenges. In this project we are proposing a cutting -edge solution that utilizes an AI model specifically YOLO (You Only Look Once), to detect crop ripeness and crop disease based on their leaves using an equipped rover with a camera and soil sensor that transfer data captured and display it on website interface providing a real-time insight into the conditions of crop. The proposed approach can significantly mitigate the issues that agriculture sector is facing, ultimately enhancing productivity and sustainability in food production.

### Background

Agriculture has been a keystone for human civilization for centuries. Evolving from primitive or subsistence farming to complex systems that feed billions. Traditional agricultural methods were born from local knowledge using basic tools, cultural and traditional everyday practices, sustaining communities for centuries. These methods and approaches often prioritize biodiversity and sustainability, relying on techniques such as crop rotation and organic fertilizers.

However, traditional agriculture approaches face huge challenges nowadays including soil degradation, climate change, and increasing population which as a result increased demands for food. Despite its strengths, traditional farming approaches are increasingly unable to meet the exponentially growing global food requirements. Issues such as nutrient depletion, deforestation, and soil erosion are prevalent as farmers strive to maintain productivity without the use of modern technology. The reliance on manual labor and simple tools limits efficiency and scalability, making it difficult to compete with industrialized agricultural practices that emphasize high yields through mechanization and chemical inputs[[1].](#_bookmark74)

Although agriculture sector is the backbone for Lebanon’s economy, contributing substantially in Lebanese Gross domestic product (GDP) reaching up to 80% of economic output [2] ;this sector faces systemic challenges ranging from water scarcity, infertile soils, inadequate access to technology, climate change pressures, and more. Thus, advancing agriculture through technology adoption is critical for Lebanon’s economic growth and sustainability which is a must.

There are various types of machines used in agriculture some of which are seeding machines that automate the sowing of seeds, ensuring optimal depth and spacing. Transplanting machines that move plants from a dense nursery growth stage to a more spread-out growth stage. Harvesting machines are designed for efficiently cutting plants from their roots. Robots equipped with sensors and end-effectors to identify and pick ripe produce. Weeding and pest control units that use precision application of herbicides

or mechanical methods to manage weeds and pests. monitoring and scouting drones or ground vehicles that gather data on crop health, moisture levels, and other critical parameters.

To address these challenges, there is a pressing need to integrate new technologies into agricultural practices. Innovative methods such as precision agriculture, which utilizes data analytics using IoT devices and sensors to optimize resource use, can significantly improve productivity while minimizing environmental impact. Technologies like drones for crop monitoring, robotic harvesters for labor-intensive tasks, and advanced irrigation systems can transform traditional farming into a more efficient and sustainable practice.

In this project we seek to demonstrate a proof-of-concept precision agriculture solution using robotic systems and sensors. The project aims to develop a rover equipped sensing for soil health analytics and 2 AI models for computer vision that can traverse farmlands and gather agricultural insights without needing a large workforce.

### 2.2- Objectives:

The objectives of this project are as follows:

* Promote sustainable farming practices to reduce soil erosion
* Increase Agricultural yield
* Offer remote monitoring to crop
* Automate collected sensor data
* Achieve real-time detection and classification.
* Detect with low latency

### 2.3- Motivation :

The recent escalation of conflict in Lebanon-Israeli War 2024, has had vast devastating impacts on the agricultural sector in specific and the environment in Lebanon in general . This war resulted in over 10,800 hectares of land stated Dr.Hadi Jaafar professor and chair of Department of Agriculture in American University of Beirut ,where areas four times larger than Beirut ,burned and significantly damaging crops and irrigation infrastructure, increasing the need for innovative solutions to restore and enhance the agricultural landscape is more urgent than ever. The fires impacted primarily shrubland and grassland while also inflicting extensive damage on forested areas along the 120km southern border, penetrating up to 10km inwards in some areas [[3].](#_bookmark74)

The cornerstone of Lebanon’s agriculture economy ,olive harvest with approximately 25% of agriculture lands are dedicated to olive groves primarily in south which due to the conflict it has been severely damaged .Lebanon and southern regions are known for producing the highest quality of olive oil globally Lebanon exported $ 30million USD worth olive oil in 2016 (EU-funded study ) .



*Figure 1 Burnt Areas in South Lebanon(October 2024 assessment ) -Data Source :European Forest Fire Information System*

In addition to agriculture in the Beqaa Valley the war led to damage to this crucial agriculture region disrupting export routes jeopardizing potato harvests such that Lebanon exports over $20 million USD worth annually mainly from Beqaa region in addition to cereal production with over 10,000 ha planted annually producing more than 40,000 ton of wheat and this disruption eventually impact the live hood of both farmers and market stability .

The approach proposed in presents an opportunity to address both immediate challenges posed by recent conflicts and long-term sustainability goals in Lebanon's agriculture sector. By utilizing advanced technologies for land monitoring that could help in damage assessment, crop monitoring, and precision agriculture, this initiative aims to not only restore but also enhance the resilience and productivity of Lebanon’s agricultural landscape in the wake of devastation supporting Lebanon's agriculture sector recovery.

### 2.4- Applications:

In this section, we will state the different settings and applications where the rover robot can be effectively deployed, highlighting the importance of applying this advanced technology in those applications, revolutionizing the traditional farming practices, thus ultimately contributing to a more efficient future in agriculture. The following is a list of applications where our rover can be implemented:

* Large Farms: enables farms to make the most out of their resources by constantly monitoring soil quality, fruit ripeness, and crop health real-time. By providing necessary and important data eventually increasing crop yield fertilization, and reduced wastage, and reduced operational costs. This, in turn, leads to more robust crops and higher productivity. Recent years show a significant growth in studying and inspecting how crops are grown and monitored in large field areas using autonomous techniques their operations and reducing environmental impact.
* Small Tight Farms: Numerous problems and constraints face farming systems nowadays which include small land sizes, unorganized crops, the decrease in the number of resources, and the increase in the degradation of quality of soil limits crop production and threatens overall food security. Different climate change circumstances such as frequent and extreme weather events and conditions magnify these problems Implementing this approach in smallholder farms represents a promising solution to the challenges these farms encounter. Our proposed approach offers real-time insights about soil conditions and crop health, and ripeness. Additionally, it increases the aware of small farm holders of plant conditions by providing soil data from sensor, which enables them to take necessary measures. Our approach provides tailored solutions to the specific challenges. Encountered by smallholder farms, ultimately enhancing crop production, food security, and the overall well-being of smallholder farmers.
* Greenhouses: advanced technologies integration, connected devices, and data- driven solutions in smart greenhouses can automatically monitor temperature, humidity, soil moisture, and ripeness of existing plants. This enables precise resource management, conserves water and nutrients, and ultimately contributes to cost reduction. Overall, applying such rover in greenhouses fits with modern agricultural practices and offers advantages to the quality of crops, efficiency of resources and increasing profitability.
* Vertical Farms: The use of this project, utilizing state of the art technology, including sensors, analysis of data, and automation maximizes the ability to adjust environmental variables such as light, temperature, humidity, and nutrient levels. This approach that allows optimal conditions combined to the specific needs of each crop results in an accelerated growth and enhanced productivity. According to a study done in, vertical farming has a huge potential in sustaining the production of food and all related services in urban and crowded areas, ultimately contributing to sustainable and economically viable agricultural practices.

While there are various applications for implementing this project, the ones highlighted above hold particular reference in today's modern problems of agricultural landscapes. Development and innovation are urgently needed in a world where the population continues to grow to sustain the daily increasing needs of supplies, ablate the amount of food being wasted, and supply access to food for all people especially those who are suffering from hunger and malnutrition. However, the potential use for this system extends far more than just its current applications. This system shows a huge promise in revolutionizing traditional farming methods by ensuring and enabling sustainable practices. Ongoing research and development indicate a growing range in the potential uses that could optimize agricultural practices in the future.

## Literature Review

### Introduction:

The integration of robotic technology in agriculture has been a growing area of interest in recent years. This integration has emerged as a solution to address the increasing demands for efficient food production. Recently, the evolution and progress in robotics and AI enabled the development high-end rovers equipped with sophisticated sensors and computer vision systems that provide some insights whenever we want wherever we are. In this chapter we will explore some of the latest conducted of research on robotic rovers in agriculture, focusing on their capabilities in soil data collection and computer vision in this field.

### Existing Work

In 2024, Tzani et al [[4]](#_bookmark74), investigated in assessing fruit quality and emphasizing its importance on agriculture sector and its effect on producers ,distributors ,consumers and countries economy .Stating that Artificial intelligence can aid and assess the quality of the fruits using images captured .The author presents a Deep learning model specifically vision transformers (ViT) to evaluated fruits using feature extraction .The ViT model proposed is trained on diverse set of fruit datasets enabling it to differentiate between rotten and ripe or good fruit based on its appearance rather than predefined quality attributes .The model achieved a high including the following : 99.50% percent accuracy for apples , 99% percent for cucumbers, and lastly 100% for grapes. Other fruits including the following: kakis, oranges, and tomatoes also resulted in high accuracy rates ranging from 98% to 99.50%. However, the model exhibited slightly lower performance in identifying guavas, lemons, limes, mangoes, pears, and pomegranates, with accuracies around 97% to 97.50%. The proposed approach solved an important problem that usually in traditional agriculture methods required extensive manual inspection. The ViT model proposed that abled automating fruit quality offered a scalable solution that can benefit various sectors.

In 2023, Li et al [[5]](#_bookmark74) , Li et al. indicated that the rapid development of computer technology has greatly facilitated international agricultural modernization and improved the efficiency of agricultural production. In this respect, Li et al. have suggested the Strawberry R-CNN, a new model proposed especially for the intelligent identification and counting of strawberries within their natural environment. The model proposed is built based on a refined existing model, namely the Faster R-CNN architecture, through several key modifications aimed at improving recognition accuracy. The original VGG16 backbone in Faster R-CNN was replaced by an improved multi-cascade network structure for the extraction of features, making it more capable of obtaining rich location data and fine details often lost in their higher-level abstractions. The experimental results of the proposed model showed that the mean precision of the Strawberry R-CNN model is 0.9019 for ripe strawberries and 0.8447 for immature, or in other words, unripe strawberries, with a mean average precision (mAP) of 0.8733. this approach underlines the potentials of advanced deep learning techniques for improvement in agricultural practices by enhancing automation and precision in crop management.

In 2024, Zhao et al [[6]](#_bookmark74) , conducted an approach using a lightweight version of YOLOv5 algorithm specifically YOLO-Granada , authors used this model to detect pomegranate fruit using an intelligent management systems for pomegranates orchards seeking to improve yields and address labor shortages .The paper states that currently most solutions use deep learning to detect pomegranate however deep learning is not effective in detecting small targets and large parameters , and it has slow computational speed so they suggested using YOLOv5 and for the pomegranate feature extraction they utilized a lightweight ShuffleNetv2 network .The results showed that the YOLO- Granada reached 92.2% accuracy percentage which is slightly lower than that in original YOLOv5 model which is 92.9% .On the other Hand YOLO-Granada achieved 17.3% increase in detection speed after compressing model parameters ,floating -points operations and overall size to 54.7% ,51.3% and 56.4% from the original network respectively .

In addition, the approach used showed a real time capability by processing 8.66 images per second. Moreover, the authors explored the development of an Android based application utilizing Nihui convolution neural network framework for detecting pomegranate real -time.

In 2023 Wang et al [[7]](#_bookmark74) , proposed an improved target detection algorithm based on YOLO v5n using K-means++ clustering algorithm to update the scale and aspect ratio of the anchor box so it can adapt to cherry tomatoes shape .Then they used coordinate attention mechanism (CA) to expand receptive field range reducing the interference that the model may face due cluttered background such as dead leaves , branches ,..etc that may affect the recognition of cherry tomato maturity. After that they replaced the traditional loss function with bounding box regression loss with dynamic focusing mechanism (WIoU) loss function. To address the boundary box regression balance problem between high-quality and low-quality data the authors introduced outlier degree and dynamic nomontonic mechanism. The results of the provided approach indicated that the improved model achieved a 1.4% increase in both recall and precision with respect to other YOLO models also the model achieved average accuracy mAP of 95.2% and average detection time of time 5.3ms which makes it highly suitable for deployment in embedded systems and mobile devices. The proposed approach provided rapid and accurate target recognition for cherry tomatoes.

In 2023 Pickett et al [[8]](#_bookmark74) , conducted a comparative between unmanned aircraft systems and agro-terrestrial (ground-based) surveying ,in this paper the author’s compared the accuracy, precision, time, and cost efficiency of using a small unmanned aircraft system (sUAS) for aerial surveying versus traditional ground-based (agro-terrestrial) surveying methods in an agricultural field. the researchers conducted both aerial and ground-based surveys on a 14-hectare field in Arkansas. concerning the aerial surveying they used a DJI matrice 300 RTK sUAS with a 45-megapixel camera. while the ground-based survey they used a utility vehicle equipped with a Trimble r8s GNSS receiver, collecting data at three different track spacings (7.62 m, 15.24 m, and 30.48 m). they also used statistical methods to compare the elevation data from the two approaches.

After conduction different experiments on different track spacings the author’s states that ground surveying is more widely adopted and requires less technical expertise with high accuracy in ground vegetation and low processing time also it can survey at night unlike aerial surveying .In addition ground-based surveys is more time-consuming, however, the ground-based surveys provided more detailed and accurate measurements. The authors suggest that ground-based approaches could be better than sUAS for applications that require higher accuracy, such as formal land surveys, as professional land surveyors are typically interested in absolute accuracy.

In 2024 Ozkan et al [[9]](#_bookmark74) , discusses the use of drones stating that it’s relatively new technology in agriculture and that there is limited amount of credible published research data evaluating the performance of drones compared to ground-based approaches providing the challenges and limitations in utilizing this method .Stating some of its limitations such as the frequent need to charge covering few acres per hour compared to ground sprayer for pesticides .In addition the author demonstrates the significant challenge of using drones due to it’s limited weight requirement stating FAA restrictions on drones, such as: “a drone must weigh 55 pounds or less including its payload” and drones can be flown only from 30 minutes before sunrise to 30 minutes after sunset while ground-based approaches can be used at night .

In 2024 Das et al [[10]](#_bookmark74) , proposed an agriculture rover to address the growing demand for sustainable agriculture practices .The agriculture rover is used for soil analysis and YOLOv5 model for detecting tomato ripeness .The soil analysis was done using NPK (Nitrogen ,Phosphorus and Potassium ) sensor which can give farmers insight about soil health and take necessary measures accordingly .The authors used dataset containing 500 images taken from tomato field and divided them accordingly : 350 images for training set , 100 for validation ,and 50 for testing set .

The model approach proposed resulted in 0.8518% for precision and 0.7624% recall providing a real demonstration for successful integration of computer vision into agriculture rover for crop insights and soil analysis conducting extensive field trails to assess efficiency of the rover in crop detection .

In 2020 Rajendran et al [[11]](#_bookmark74) , proposed IoT and AI integration to combat farmers agricultural loss due to plant disease ensuring there need for early pant disease identification mechanism .The observation for each and every individual plants in the farm for detecting early signs of diseases is labor intensive and time consuming .The authors utilized rover equipped with a camera and gps module to capture images of plant leaves through farms and greenhouses .The dataset employed is “Plant Village” plant disease was prepared by Hughes containing 60,000 images of more than 35 diseases of 16 plant species some of which are Pepper, grapes strawberries the four diseases that theses species suffer from are Black rot and Black measles in grapes , bacteria spot in pepper and leaf scorch in strawberry .The 2 deep learning models used in this approach are VGG16 and InceptionResNetV2 .The approach proposed resulted in 97.56 % training accuracy and 93.21 % for validation accuracy using VGG16 while in InceptionResNetV2 the training accuracy reached 98.32 % and validation accuracy reached 95.24% it’s worth noting that number of parameters in VGG16 is 134,289,223 Unlike InceptionResNetV2 having 58,091,591 having less than the half of the parameters in VGG16 .

In 2024 Karim et al [[12]](#_bookmark74) , integrated an edge device namely Nvidia Jetson Nano that can be utilized in rover with Deep learning model to leverage crop yield and detect crop disease precisely and utilized a python GUI (PyQt5) to interface the collected data .The proposed approach involved the use of modified MobileNetV3Large to detect crop disease early also the author performed a comparative analysis comparing the performance of the purposed approach some of which are :MobileNetv3Small

,DenseNet21 ,EfficientnetV2B1 .The proposed approach reached the highest training and test accuracies of 99.66% and 99.42% outperforming previously mentioned models.

In 2024 Ahmed et al [[13]](#_bookmark74) ,investigated the performance of two lightweight object detection models in the application of deep learning for precise tomato disease detection focusing on four categories : healthy , splitting rotation , sun-scaled rotation and blossom end rot .The two dataset were compared on custom tomato disease dataset .Initially the authors trained both models without data augmentation to establish a baseline and then utilized diverse data augmentation techniques from Roboflow to expand the dataset content .After data augmentation both models were re-trained and all disease categories were analyzed .The results showed that the models had significant improvement in accuracy after data augmentation .In addition the results showed that YOLOv8l reached 79.2% precision 70.1% recall 78.9% mAP50 55% mAP50-95 achieving higher accuracy compared to YOLOv5l which reached only 69.8% precision 62.3% recall 67.9% mAP50 49.5 mAP50-95 before data augmentation . Furthermore, the results improved after data augmentation such that YOLOv8l 91.6% precision 83.1% recall 88.5% mAP50 60% mAP50-95 while YOLOv5l only reached 89.3% precision 74.4% recall 85.2% mAP50 58.5% mAP50-95 .The results indicates that YOLOv8l outperforms YOLOv5l.

In 2024, Mac et al [[14]](#_bookmark74) , investigated the use of soft computing methods on autonomous intelligent agriculture particularly on systems for autonomous greenhouse navigation integrating fuzzy control algorithm with deep learning based models for classification that identifies illnesses in tomato plants through images of their leaves .This paper utilizes upgraded Deep Convolutional Generative Adversarial Network (DCGAN) which generates augmented images of diseased tomato leaves from the original samples there by enhancing the training dataset significantly .To identify which deep learning model is the optimal one in this application the author did a comparative analysis between various deep learning models some of which : VGG19, Inception-v3, DenseNet-201, and ResNet-152 .The approach proposed resulted in the following accuracies : 92.32%, 90.83%, 96.61%, and 97.07%, respectively using “original Plant Village “ dataset. The use of augmented dataset created by the proposed DCGAN, significantly increased the accuracy of ResNet-152 model reaching accuracy of 99.69% after it was 97.07%. The proposed approach increased the accuracy of DCGAN favoring the performance of deep learning models for monitoring greenhouses plants and detecting diseases.

### Summary Table:

*Table 1: State of the Art Research Comparison*

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Authors | Year | Method Used and Outcomes |
| A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features | Tzani et al | 2024 | Deep learning model based on ViT with the purpose of assessing quality of fruits using images .The results showed the following accuracies: 99.50% for apples, 99% for cucumbers, and 100% for grapes, while other fruits like kakis, oranges, and tomatoes had high accuracy rates that ranged from 98% to 99.50%. It showed a bit of a lower performance for guavas, lemons, limes, mangoes, pears, and pomegranates, with accuracies ranging between 97%  and 97.50%. |
| Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R- CNN | Li et al | 2024 | The model is Strawberry R-CNN, extending the architecture of Faster R-CNN that performs the counting and intelligent recognition of strawberries The experimental results showed that Strawberry R- CNN achieved an mAP of 0.9019 for ripe strawberries and 0.8447 for immature ones, with a mAP of 0.8733, and counting accuracies of 99.1% for  ripe and 73.7% for immature strawberries. |
| YOLO-Granada: a lightweight attention Yolo for pomegranates fruit detection | Zhao et al | 2024 | YOLOv5-Granada pomegranate detection in an intelligent management system of a pomegranate orchard. They results reported that YOLO-Granada achieves a slightly reduced accuracy compared with the original YOLOv5 model-92.2% versus 92.9%. model parameters compression, floating-point operation compression, and size, respectively, to 54.7%, 51.3%, and 56.4%, increased detection speed by 17.3% for the YOLO-Granada model |
| A Lightweight Cherry Tomato Maturity Real-Time Detection Algorithm Based on Improved YOLOV5n | Wang et al | 2023 | The target detection algorithm used is based on YOLO v5n using K-means++ clustering algorithm , The results showed that the improved model outperformed other YOLOv5 model by increasing recall and precision by up to 1.4% also achieves an average accuracy mAP of 95.2%, with an average detection time of time 5.3ms, hence very suitable to be deployed in embedded systems and mobile devices. |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Authors | Year | Method Used and Outcomes |
| Small Unmanned Aircraft Systems and Agro-  Terrestrial Surveys Comparison for Generating Digital Elevation Surfaces for Irrigation and Precision Grading | Pickett et al | 2023 | The aerial surveying, they used a DJI matrice 300 RTK sUAS with a 45-megapixel camera. while the ground-based survey they used a utility vehicle equipped with a Trimble r8s GNSS receiver, collecting data at three different track spacings (7.62 m, 15.24 m, and 30.48 m).The results showed that ground surveying is preferred due to its high accuracy in measuring ground vegetation, lower processing times, and the ability to conduct surveys at night. While ground-based surveys are more time- consuming, they provide greater detail and precision |
| Designing and  development of agricultural rovers for vegetable harvesting and soil analysis | Das et al | 2024 | Agriculture rover is used for soil analysis and YOLOv5 model for detecting tomato ripeness. The model approach proposed resulted in 0.8518% for precision and 0.7624% recall. |
| Drones for  Spraying Pesticides Opportunities and Challenges | Ozkan et al | 2024 | The author discusses the use of drones stating some of its limitations such as the frequent need to charge covering few acres per hour compared to ground sprayer for pesticides and its limited weight requirement stating FAA restrictions on drones, such as: “a drone must weigh 55 pounds or less including its payload” and drones can be flown only from 30 minutes before sunrise to 30 minutes after sunset while ground-based approaches can be used at night |
| Automated Plant Disease Detection using Deep Learning Architectures with Autonomous rover | Rajendran et al | 2020 | The rover equipped with a camera and gps module to capture images of plant leaves through farms and greenhouses was used with 2 deep learning models VGG16 and InceptionResNetV2 .The results showed  97.56 % in training accuracy and 93.21 % for validation accuracy using VGG16 while in InceptionResNetV2 the training accuracy reached  98.32 % |
| Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM | Karim et al | 2024 | Nvidia Jetson Nano edge device was used to detect crop disease precisely and utilized a python GUI (PyQt5) to interface the collected data , the model used is modified MobileNetV3Large to detect crop disease early. The results were compared with MobileNetv3Small,DenseNet21,EfficientnetV2B1and the proposed method reached the highest training and test accuracies of 99.66% and 99.42% outperforming |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Authors | Year | Method Used and Outcomes |
| Improved Tomato Disease Detection with YOLOv5 and YOLOv8 | Ahmed et al | 2024 | Rover equipped with a camera and gps module was used to capture images of plant leaves through farms and greenhouses, farm for detecting early signs of diseases the results showed after data augmentation such that YOLOv8l 91.6% precision 83.1% recall 88.5% mAP50 60% mAP50-95 while YOLOv5l  only reached 89.3% precision 74.4% recall 85.2% mAP50 58.5% mAP50-95 .The results indicates that YOLOv8l outperforms YOLOv5l. |
| Intelligent agricultural robotic detection system for greenhouse tomato leaf diseases using soft computing techniques and deep learning | Mac et al | 2024 | Deep Convolutional Generative is used with the following models VGG19, Inception-v3, DenseNet- 201, and ResNet-152 to detect tomato disease. The results showed the following accuracies : 92.32%, 90.83%, 96.61%, and 97.07% and after using DCGAN, significantly increased the accuracy of ResNet-152 model reaching accuracy of 99.69% after it was 97.07%. |

### Alternative Design :

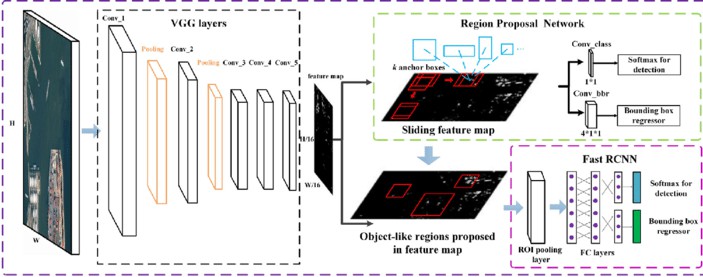
In this section, the different design approaches and tools that could be adopted for our project in relation to the 3 phases, which are divided as follows (phase 1: AI model development, phase 2: rover design and prototyping, and phase 3: website development) are going to be explored. This may enhance the effectiveness of our project by exploring different or various design approaches and tools and ensuring that they align with our objectives. This section focuses on the illustration of different methodologies and technologies that can be embedded into our project framework, making it flexible for adaptation and optimization of strategies. We have to make sure that with the choice of different options, our final design will satisfy not only the technical needs but also the practical needs of farmers and landowners.

#### Concerning the AI Model :

* + 1. Faster R-CNN :

Faster region-convolution neural network is two stage object detection model that is built upon previous models R-CNN .This model combines the benefit of deep learning

,convolution neural networks and region proposal network (RPN) that automatically generate the region proposal at the first .The Architecture of Faster R-CNN is divided into backbone networks that are considered as feature extractors commonly it included a pretrained Convolution Networks like shown in figure 3 below :



*Figure 2 Faster R-CNN Architecture [17]*

Where in the architecture presented above VGG was used from which the input image is processed to generate a feature map that captures hierarchical representations of visual information.

The feature map is smaller than the input image, though it retains essential semantic information, which is important for both region proposal and object classification tasks.

Once the region proposals are generated , the results are inputted to an ROI pooling layer which is used to extract a fixed length of vectors for each of the proposals Then the vectors are fed into two parallel fully connected layers which are divided as follows : one for object classification and the other is used to refine the bounding box coordinate While it’s had good characteristics in object detection it can underperform compared to other architectures as shown in table 2 below:

*Table 2 Yolov8 and Faster RCNN Comparative Analysis[18]*

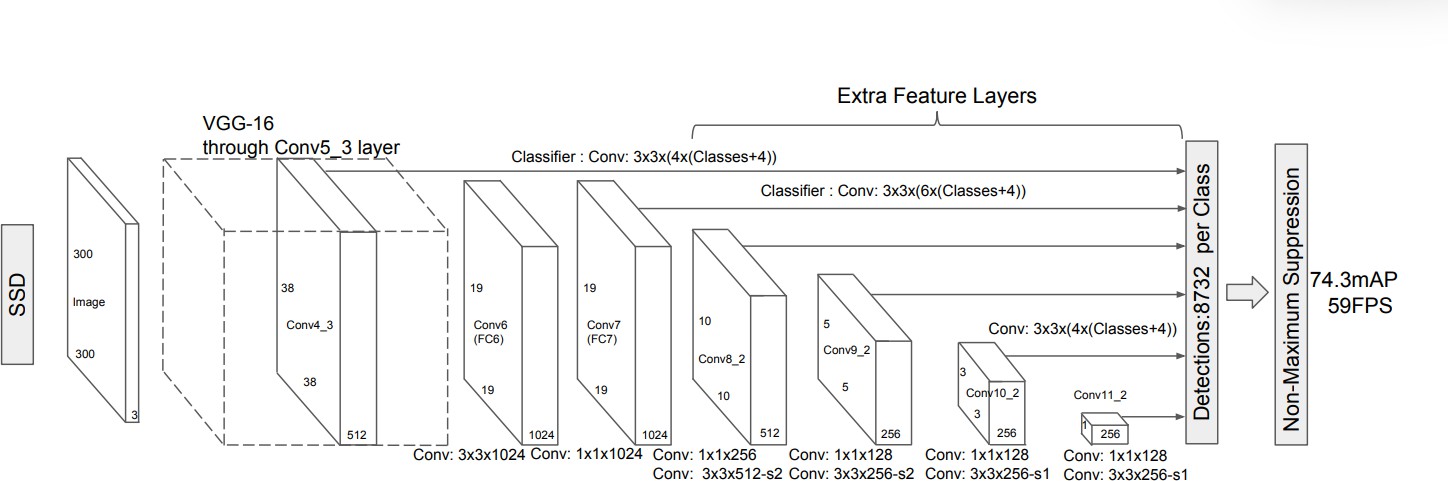
|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | mAP@50 | GPU latency (ms) | Customization |
| YOLOv8 | 0.62 | 1.3 | Customizable |
| Faster R-CNN | 0.41 | 54 | Limited |

Although Faster RCNN offers efficiency and accuracy in detecting objects yet YOLO provides lower latency approach as shown in table 2 where YOLOv8 reached 1.3 ms whereas Faster R-CNN needed 54ms greater than that offered by YOLO by more than 50 times one of the reasons that YOLO architecture achieves lower latency is due to the fact that YOLO is one stage while Faster R-CNN is two stage architecture . Despite the fact that in our project low latency for faster detection is important , accurate detection is important too and after comparing the latencies of both architectures we compared mAP@50 which is mean Average Precision that measures the precision and recall of a model at an Intersection over Union (IoU) threshold of 0.50 the YOLO architecture provided higher value reaching 0.62 compared to Faster R-CNN which reached only

0.41 which implies that YOLO architecture suits our proposed project and complies with our objective compared to Faster RCNN.

* + 1. SSD :

Single shot MultiBox Detector is single -stage object detection model unlike Faster RCNN which implies that SSD identifies targeted objects in an image after single forward pass of the network. The SSD discretize the output space of bounding boxes into a set of default ones but with different ratio aspects and scales at each feature map location enabling the model to predict bounding box offsets and class scores simultaneously for multiple object having various sizes. As previously explained model SSD consists of previously trained CNN such as VGG16 present in figure 4 below :



*Figure 3 Single Shot Mutibox Detection Architecture[19]*

which is used to processes an input image and generate feature maps. In addition, further convolutional layers are involved, reducing their size successively thereby allowing the model to detect of objects ranging from small to big by size. A key feature of SSD is its use of multi-scale feature maps; it uses the output from various layers of the base network to capture information across multiple resolutions which improves the detection of objects of different sizes. For every default box that is generated at each feature map location, SSD predicts the class confidences-the probabilities that each box contains some specific object class-and the bounding box adjustments to refine the position and size of these boxes for a better alignment with the objects detected. The SSD combines the localization loss, which was measuring exactly how well our predicted bounding boxes match ground truth, with the confidence loss assessing the accuracy in class prediction. At the end SSD uses Non-Maximum Suppression (NMS) which is used to remove redundant boxes to retain just the most confident predictions that finalize detections.

While SSD is a single-stage object detection which have lower inference with respect to other multi-stage object detection models yet it has some pitfalls that prevent it from being the optimum model for this project the model will be compared according to performance and accuracy.

Where Kaliappan examined the performance of both models YOLOv8 and SSD for precision poultry farming management and the results are shown below in Table 3 :

*Table 3 YOLOv8 versus SSD comparative analysis[20]*

|  |  |  |
| --- | --- | --- |
| Metrics | YOLOv8 | SSD |
| Precision | 0.9677 | 0.89 |
| Recall | 1.0 | 0.65 |
| mAP@50 | 0.987 | 0.77 |

While both YOLO and SSD showed nearly similar precision yet YOLO outperforms SSD recall and in mAP@50 mean Average Precision that measures the precision and recall of a model at an Intersection over Union (IoU) threshold of 0.50 which makes SSD have low accuracy compared to YOLOv8 model .The tradeoff between speed and accuracy of SSD makes it less efficient for our application in this project were precise detection is required to detect small crops or subtle signs of disease on leaves

Now concerning the performance of SSD model , YOLO v8 can process up to 155 Frames per second While SSD can only reach at most 46 frame per second as shown in in table 4 below:

*Table 4 Performance comparison between YOLOv8 and SSD[21]*

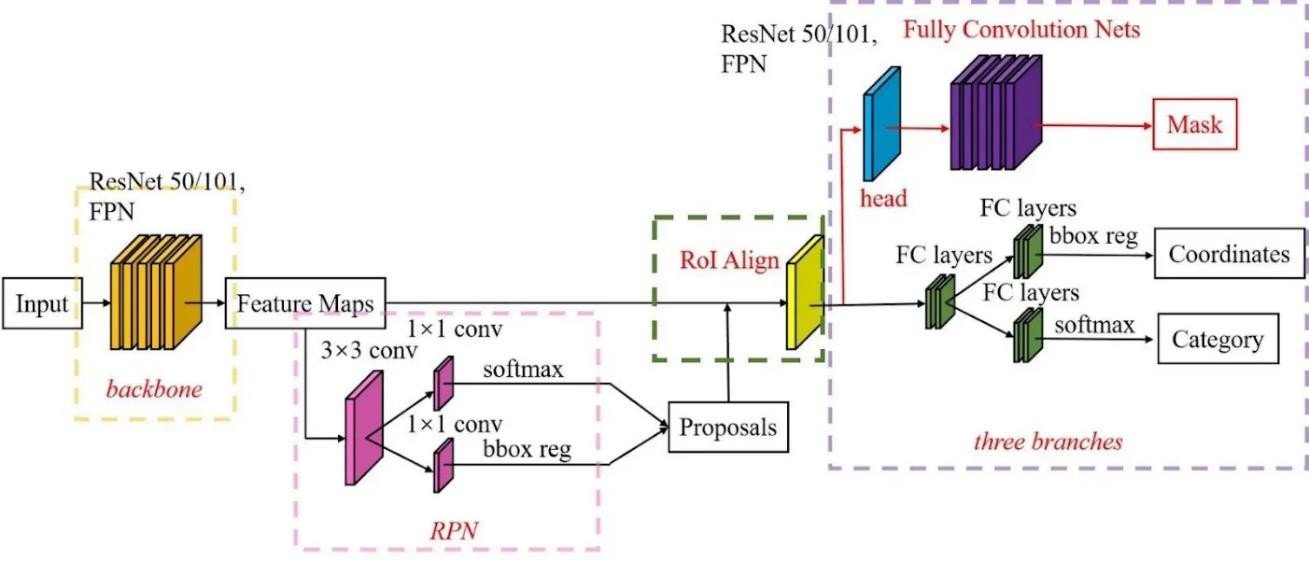
|  |  |
| --- | --- |
| Model | Frames Per Second (FPS) |
| YOLOv8 | 40-155 |
| SSD | 22-46 |

The low accuracy and low real time processing speed of SSD compared to YOLOv8 prevent us from choosing SSD such that other architectures offer higher accuracy and higher processing speed reaching 155 FPS .This implies that SSD is not suitable for real-time crop health monitoring making it unsuitable for our project .

* + 1. Mask R-CNN :

Mask region convolution neural network is an extension of Faster region convolution neural network object detection used for instance segmentation and object detection.

This model can detect an object in an image and generate high-quality segmentation masks for every detected instance. This allows the delineation of object boundaries at the pixel level. Starting with its backbone which is usually a pre-trained network like RestNet50 shown in Figure 4 below :



*Figure 4 Mask R-CNN Architecture[22]*

Which is used to process the input image for high-level features.To handle objects of different sizes effectively, an ( FPN ) Feature Pyramid Network is incorporated that builds a multi-scale feature pyramid by providing features in different resolutions.Then, the Region Proposal Network generates region proposals by sliding on the feature map to predict bounding boxes and object scores for potential objects in an image, outlining regions of interest. Instead of using traditional ROI Pooling or in other words Region of interest pooling, Mask R-CNN uses Region of interest Align (ROIAlign) , which relies on bilinear interpolation to ensure the exact alignment of the extracted features with the proposed regions which enables it to avoid misalignment and increases the segmentation accuracy. Mask R-CNN extends this further by adding a mask branch parallel to the existing branches for classification and regression of bounding boxes, by predicting a binary mask for each proposed region to outline the exact pixel-level boundaries of detected objects. The final output from the network would include a bounding box, class labels, and pixel-wise masks for every detected object allowing instance segmentation with great details combined with the usual object detection.

While Mask R-CNN is a extended version of faster R-CNN and its capable of object detection and instance segmentation yet it has some pitfalls that prevent it from being the optimum model for this project the model will be compared according to performance and accuracy

Sapkota in his attempt to compare between YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environment to segment apple fruitlet ( unripe or immature fruit) resulted in the following :

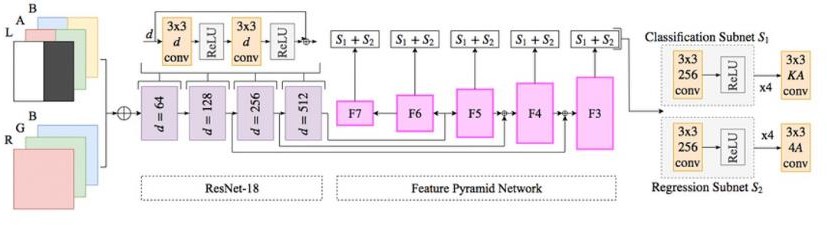
*Table 5 Comparative Analysis between Mask R-CNN and YOLOv8[23]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | mAP @0.5 | Inference Time(ms) | Frames Per Second(FPS) |
| YOLOv8(single -class) | 92.9 | 97 | 0.902 | 7.8 | 128.21 |
| Mask R-CNN(Single-Class) | 84.7 | 88 | 0.85 | 12.8 | 78.13 |
| YOLOv8(Multi-class) | 90.6 | 95 | 0.74 | 10.9 | 91.74 |
| Mask R-CNN(Multi-Class) | 81.3 | 83.7 | 0.700 | 15.6 | 64.10 |

Where YOLOv8 outperformed Mask R-CNN in both multi and single class dataset .in addition YOLOv8 had higher precision , recall and mean average performance at 0.5 threshold with lower inference time and more real time processing speed reaching in this example 128.21FPS which implies that Mask R-CNN is not the optimum model for our application that have similar objective to the above experiment .

* + 1. RetinaNet :

Retina Net is name is inspired by its ability to detect objects with high accuracy across a range of scales, mimicking human retina and how it perceives visual information .It’s a one-stage object detection model ,designed to effectively handle the challenges of imbalanced data and varying object sizes through its architecture that includes the following : a backbone network, a Feature Pyramid Network (FPN), and two task- specific subnetworks for classification and regression. The backbone network, usually Rest Net similar to one presented in figure 6 below :



*Figure 5 RentinaNet Architecture[24]*

Where RestNet-18 that is used to extracts feature maps from the input image at different scales. The Feature Pyramid Network (FPN) enhances this by creating a multi-scale feature pyramid that allowing the model to detect objects of various sizes; it does this through a top-down pathway that up samples higher-resolution feature maps and lateral connections that merge these with lower-resolution maps. The classification subnetwork predicts the probability of an object being present at each spatial location for each anchor box, while the regression subnetwork refines the bounding box coordinates. A key innovation in RetinaNet is its use of Focal Loss, which addresses class imbalance by focusing more on hard-to-detect objects, thereby improving overall detection performance. The output of RetinaNet consists of bounding boxes and class probabilities for detected objects, making it effective in various applications, particularly those involving dense or small-scale objects.

Although it provides significant features the inference time compared to YOLO models is slow and the reliance on anchor boxes can complicate the dedication of overlapping object that are found in crop fields [[25]](#_bookmark74)

#### Concerning the Website Development :

In this section, we will look at the various design alternatives for the website tools of our project, including the basic blocks of the frontend, backend, communication protocols, database, and cloud platform. We considered several frontend frameworks for user interface, evaluated different backend solutions to handle data robustly, and explored different communication protocols for dependable data exchange between the rover and the web application. Further, database systems were analyzed for the performance of storing and retrieving data, and cloud platforms were analyzed in terms of their scalability and integration capabilities. This in-depth exploration shall help identify the best technology combination to enhance system performance and the user experience

#### Frontend Development

Frontend development includes the graphics face along with the overall design aimed at customers in terms of experience. It serves as the bridge between the users and the software features. In our web interface, the frontend will be applied to view the data from the rover and interact with the rover in a live manner; this will be important in achieving the user usability. In order to decide which front-end technologies, we should use, whether React, Vue.js, or even Angular, we considered its impact on performance, maintainability, and user satisfaction. We evaluate these frameworks to find which one would be most suited for the project.

React is a free popular JavaScript library whose primary aim is to create user interface for particularly single page applications. By utilizing a component-based approach, it allows developers to create scalable reusable UI elements which in turn result in better code maintenance. Given that React is popular and supported by a vast community with a variety of state management solutions like Redux and Context API it would definitely be favorable for developing an agricultural rover’s interface that requires a lot of performance.

Vue.js is a JavaScript-based framework for building user interfaces, it’s single page applications and complex web applications targeting end users. It is ranked up as one of the most learnable programming languages due to its ease of use and integration capabilities with other applications. Its template syntax is easy to understand and the data binding is a reactive data which makes management of states very easy. In our project, this would enable the swift creation of a flexible frontend making interaction with Flask based backend seamless[[29].](#_bookmark73)

Angular is resolves complex single page applications through its architectural approach based on HTML and Type script. It has the features such as dependency injection and two-way data binding but is harder to learn compared to others frameworks. But due to their strong conventions and tooling it increases productivity of bigger teams.

For our interface, it would can use to construct the orderly frameworks that can be well scaled on data streams and user inputs

*Table 6 Frontend Comparison Table [29]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature/Aspect** | **React** | **Vue.js** | **Angular** |
| **Type** | JavaScript library | Progressive JavaScript framework | Complete framework |
| **Learning Curve** | Moderate | Easy | Steeper |
| **Architecture** | Component-based | Component-based | Modular and service-oriented |
| **Performance** | High performance with virtual DOM | Great with lightweight core | Good, but can be complex for large apps |
| **Community Support** | Large community and ecosystem | Growing community, focus  on simplicity | Strong support and large ecosystem |
| **Use Cases** | Dynamic, complex UI | small to medium projects | large-scale applications |
| **Integration** | Easily integrates with various backends | Simple integration with REST and GraphQL | Well-suited for RESTful services |
| **Development Speed** | Fast development | Rapid development | Slower |

#### Backend Development

While choosing the backend framework to be adopted for the Web interface of our argi- robot project, we considered some of its specific needs, such as real-time data processing and ease of integration with AI models. Two of the key’s considered frameworks were Django and Flask.

Django is a high-level web framework intended for big and complex projects. It has integrated all the tools that a professional might need, such as an admin panel and authentication system, which by default can really speed up the development of big features. Although Django could provide its services in our present project, due to its comprehensive nature, it gives birth to unnecessary complexity within the system, making it a bit heavier than required. As our project deals with real-time data gathering and instantaneous feedback concerning soil and plant health, Django's overhead will possibly offset performance we want for seamless data exchange between the rover and the web interface[[30].](#_bookmark73)

On the other hand, Flask is a microframework quite good for small to medium projects, considering the flexibility and modularity. It lets the developer selectively integrate only those libraries that best meet the project's specific needs. In fact, Flask is a good choice in the scope of our argi-robot project, since it will be based on lightweight and speed considerations, crucial to real-time applications. The framework allows for quick exchanges between the sensors on the rover and the AI model; hence, it will be able to update data on the website at the same speed. This ensures users have timely insights concerning the condition of the soils or crops, hence giving valuable input for decision making.

*Table 7 Backend Comparison Table [30]*

|  |  |  |
| --- | --- | --- |
| **Feature/Aspect** | **Django** | **Flask** |
| **Framework Type** | Full web framework | Micro-framework |
| **Architecture** | Monolithic, with built-in components | Lightweight, modular |
| **Complexity** | High | Low |
| **Flexibility** | Less flexible | Highly flexible |
| **Performance** | Slower due to its comprehensive nature | Efficient with low latency in data transmissions |
| **Real-Time Processing** | Less suitable for applications requiring immediate feedback | Excellent for real-time applications |
| **Use Case** | applications needing extensive features | projects focused on efficiency and responsiveness |
| **Data Handling** | Slower data exchanges | Quick and efficient data handling |

#### Real time Communication:

Real-time performance in communication is crucially relevant to the success of the project due to the fact that any delay in data transmission may affect decision-making at any level of agricultural management arising from sensors and cameras timely. Three protocols were evaluated: HTTP, WebSocket, and MQTT as the primary means of communication.

#### HTPP:

The HTTP protocol follows in its design the principle of a Request-Response model. In this case, some client requests sent by an active client elicit responses from a passive web server. The classic method under which this process normally functions is the serving of HTML/CSS files to a client's browser. While HTTP works for simple data retrieval and authentication, such as user registration and login, loading of static content

like images and stylesheets, it does have significant limitations, especially with regard to our argi-robot project.

HTTP is most suitable for basic data acquisition operations and identity verification assignments throughout the project. Nevertheless, relying on HTTP alone has considerable drawbacks because HTTP is a request–response protocol, and the communication pattern is based on the request–response model, which increases the latency and overhead. This is the major limitation of HTTP presents in our project, which relies on real-time monitoring of data to efficiently manage agriculture. This will make the system slow, since refreshes the data are always late because of the request- response nature of HTTP. For example, soil levels would get updated after a long time, which could delay automated fertilizers adjustments or postpone critical diagnostics related to plant health.

#### WebSocket:

WebSocket mainly talks about how a client and a server will be able to share content through the full-duplex pipe. That is, two directions of data interchange, where the client and the server are also participating, exchange information with and within one another in turns over time. The response-method update approach of this bidirectional communication technique has several benefits over an application that needs real-time capability at a lesser cost than HTTP polling.

WebSocket can be effectively used for creating the real-time user interface on the website which is used for monitoring. Permanent connections between the server and the client provided by WebSocket let adding immediacy in updating of sensor data, video streams further improving the farmer experience during crop observation and monitoring. However, when used independently the WebSocket also poses its own difficulties, it could experience inefficiencies and yet escalate the complexity in receiving multiple data streams from the robot and other sensors; this would impact performance when expanded to more extensive agricultural systems.

#### MQTT:

MQTT (Message Queuing Telemetry Transport) is an IoT real time messaging protocol used by devices for communication. It designed to perform well under low bandwidth, high latency or unreliable networks. It uses publisher–subscriber model of messaging where one or more publishers can send messages to a broker that forwarding them to subscribers. It is well-used in IoT environment because of its efficiency and simplicity for use.

In our project, the best way to use MQTT is when receiving sensor information from the robot to a specific broker. This protocol again shines in the quick generation and dissemination of readings such as soil moisture and temperature with ability to support multiple subscribers including the monitoring website. However, one disadvantage in using MQTT by itself is the fact that some of the clients you want to carry out web- based interactions are unable to achieve two-way communication required in a personalized dashboard application. This means that although the sensor data may be transmitted efficiently the users and their interaction with the system may have to be compromised unless further additions of layers are made[[31].](#_bookmark73)

*Table 8 Communication Comparison Table [31]*

|  |  |  |
| --- | --- | --- |
| **Protocol** | **Advantages** | **Disadvantages** |
| **HTTP** | Easy to implement | Slower due to request- response model |
| Compatible with existing web technologies. | Less efficient for frequent data updates. |
| **WebSocket** | Full-duplex communication | Complicated to implement and manage. |
| Low latency which fits for real-time data transfer. | Requires-additional infrastructure to maintain persistent connections. |
| **MQTT** | Suitable for continuous data streams | May not handle large payloads |
| Lightweight |
| Great for IoT applications and sensor data. | More complex setup |
| Publish-subscribe model allows flexible  communication. |

#### Database:

The rover collects data through sensor camera streaming, which send them to website with a database solution to store and monitoring them. This approach will keep the farmer informed about the most recent crop health status and soil conditions in real time and comparing them to historical data, with a view to enhancing decision making and resource management.

This project is based on two major sources of data: sensors and cameras. The quantitative data about the agricultural environment includes soil moisture, temperature, and nutrient levels from sensor data. On the other hand, the cameras and video feeds provide visual information about the ripeness of crops, presence of diseases, and general health of the plants. The amalgamation of sensor and video data requires careful consideration of how this information will be processed and stored. For that reason, the choice between SQL and NoSQL databases becomes very important.

SQL databases, such as MySQL and PostgreSQL, are structured in how they store data. A key feature of these types of databases is that data must be inserted according to a previously predefined schema. This makes them suited for well-defined data types, such as user profiles or historical records of agricultural practices. Because they are ACID compliant, SQL databases provide substantial integrity for transactions occurring in user authentication or other critical events. However, SQL databases is restrictive in terms of

scalability and flexibility. As the volumes increase and there is a need for real-time processing of sensor data, the schema of SQL databases will prove to be a challenge.

NoSQL Databases, such as MongoDB or Cassandra, are meant for unstructured or semi- structured data and, therefore, will be best suited for the wide ranges of data generated from sensors and cameras. These databases allow for quick changes to be made in the data model with very minimal downtime, thus meeting the dynamic changing in agricultural monitoring. Another key feature of NoSQL databases is that they can scale up very quickly, and it can easily distribute data across multiple servers, something quite important as our project grows[[32].](#_bookmark73)

*Table 9 Database Comparison Table[32]*

|  |  |  |
| --- | --- | --- |
| **Feature/Aspect** | **SQL Database** | **NoSQL Database** |
| **Data Structure** | Structured Data; schema predefined. | unstructured and semi-structured data, Schema-less |
| **Sensor Data Storage** | Used for structured sensor data | Excellent towards diversified formats of sensor data. |
| **Camera-Data Storage** | The data from videos is store in metadata and take it reference | Store large volumes of video files along with metadata. |
| **Real-time-Data Handling** | Slower update for data changes very frequently | Optimal performance for frequented data changes and analytics on real-time data |

#### Cloud Platforms:

Selecting the right cloud platform and services to support the deployment and growth of the argi-robot in different markets is very crucial for the project’s success. In this section, our analysis revolves around AWS, Google Cloud Platform (GCP), and Microsoft Azure, and on their specific characteristics as far as our project is concerned Amazon Web Services (AWS) is a cloud platform that offers a variety of functionalities that are very useful for our agricultural rover project. It works well with Elastic Beanstalk for managing and scaling Flask apps, Amazon S3 to host the React UI and Amazon Document DB for the database management of MongoDB. Due to the vast range of tools offered by AWS, the integrations are seamless performances are better. Furthermore, the scalability that AWS provides enables us to convert resources on the go based on the amount of data received from the rover's sensors and cameras. Because AWS implements appropriate measures that help secure sensitive agricultural data, it is also beneficial for our project.

Google Cloud Platform (GCP) is best known for the efficient data processing, analytics and machine learning, which is very helpful for our agricultural rover project focused on data collection. Google App Engine makes the deployment of Flask backends quick with automatic scaling, while Cloud Storage helps in hosting the React UI and in datasets created by the rover. Through the integrated functionalities offered by GCP, it is possible to perform great amounts of sensor data screening for analysis and derive useful agricultural information. Thanks to the AI-powered components integrated into GCP, our app has the ability to analyze data and create accurate predictions, which streamline operations everywhere.

Microsoft Azure provides a broad range of cloud products such as analytics, computing and storage. Azure App Service makes it easy to deploy our Flask application and offers easy scalability options, while Blob Storage helps in providing scalable solution for deployment of the static files and for sensor data storage. Moreover, since Azure Cosmos DB has built-in support for MongoDB APIs, it can meet all our database management needs. The Azure IoT Hub takes added advantage of the strong integration and additional connectivity features of the rover by extending our capability to manage and analyze the data generated by the rover’s sensors. Thus, this added IoT functionality may simplify and improve our agricultural operations[[33].](#_bookmark73)

*Table 10 Cloud Platforms Comparison Table[33]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature/Aspect** | **AWS** | **Google-Cloud Platform (GCP)** | **Microsoft Azure** |
| **Core Strengths** | Different service  offerings and scalability. | data analytics and  machine learning tools. | integration with  Microsoft products and services. |
| **Deployment Services** | AWS Elastic Beanstalk for Flask, S3 for React. | Google App Engine for Flask, Cloud Storage for React. | Azure App Service for Flask, Blob Storage for React. |
| **Database Options** | Amazon Document DB (compatible with MongoDB). | MongoDB Atlas or Cloud Fire store. | Azure Cosmos DB (supports MongoDB API). |
| **Scalability** | Highly scalable infrastructure | Easily scalable for various workloads. | Scalable solutions suitable for enterprise applications. |
| **Pricing** | Pay-as-you-go. Free tier available with limited usage (e.g., 1 million IoT messages/month) Pricing varies | Pay-as-you-go. Free tier with limited resources (e.g., 1 GB storage, 100 MB BigQuery queries/month).  Pricing varies | Pay-as-you-go. Free tier with limited usage. Pricing varies based on services used |
| **Security** | Strong security features | Robust security protocols | Different security features |
| **Best Use Case** | Good for full-stack applications needing many  services integration. | Ideal for projects needs advanced analytics and  machine learning. | Best for applications with strong Microsoft  integration needs. |

.

#### Concerning the Rover Design and Prototyping :

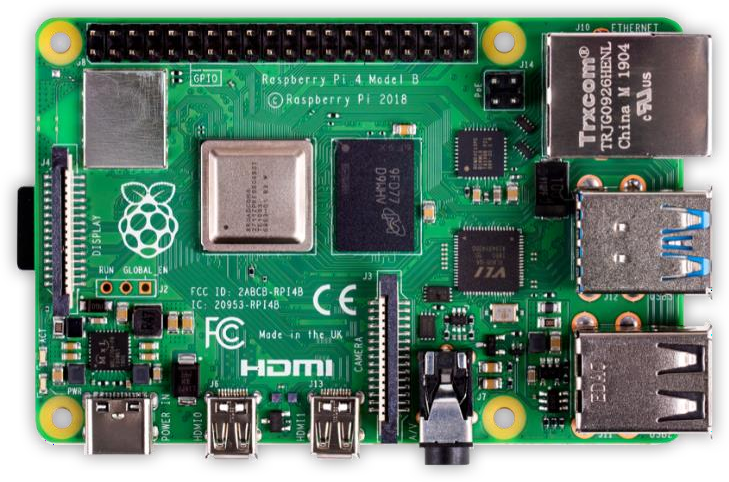
To ensure the best performance and adaptability to our project, we explored multiple options for controllers, cameras, sensors, motors, wheels, and relays. We carried out a detailed study of different controllers. Furthermore, we conducted an analysis of the potential of each of the camera options to provide quality images for AI analysis. We explored several sensors which in turn were mechanisms used for soil monitoring. We select motors that are both torque and efficient while we considered multiple wheel designs. These aspects were intended to optimize the rover's performance and adaptability to crop production systems.

### Controllers:

Controllers are key elements of every branch connected to our rover, which controls the flow of information and communication between hardware resources, including sensors, cameras. In this project, we analysis different controller’s type: Raspberry Pi, Arduino, and ESP modules. Each kind of controller has unique characteristics. The controllers provide easy and efficient communication between rover parts, thereby significantly enhancing real-time data collecting, feedback activities, and consequently quick and sure responses that are key to effective agricultural monitoring and decision-making.

Our target is to select a controller that can manage the data obtain from sensor reading and send it to website to analysis and store it. To enhance the rover’s functionality, we are going to select a controller that can execute motor functions for both the movement of wheels and deploying sensors into the soil. The controller should process data coming for camera and send it to the website for ai model analysis or the model will be embedded in the controller itself. It should facilitate real-time monitoring and analysis which will enhance the rover performanc

Raspberry Pi

Raspberry Pi 4 Model B is considered a powerful computer, providing significant upgrades in performance over others models with different variants of 1GB, 2GB, 4GB, and 8GB of RAM, and the support of dual-band Wi-Fi with frequencies of 2.4 GHz and 5 GH, and support pf Bluetooth 5. It has 40 GPIO pins for interfacing many different sensors and devices—everything on a single board capable of carrying strong computing and connectivity needs.

*Figure 6 Raspberry Pi 4 Model B*

For our project, Raspberry Pi 4 can be used in data collection and processing by connecting different sensors. It can serve as a web server for the visualization of real- time sensor data and remote supervision and control. It also has high resolution camera modules that help in taking pictures of crops for visual examination in health checks and pest infestation analysis. More importantly, Raspberry Pi can support the integration of ai model for predictive analytics and automation tasks like detect crop disease, thereby vastly improving the capability and effectiveness of the rover in agricultural management.

ESP32

The ESP32 is a very powerful microcontroller, well known for its built-in Wi-Fi and Bluetooth capabilities, thus mainly used in IoT applications. The ESP32 has a dual-core processor running up to 240 MHz and many GPIO pins, so it really can handle different tasks quite efficiently. The npk sensor are part of a project that needs to gather real-time environmental data. These kinds of data can be locally processed or sent to the cloud for monitoring and analysis. This project allows for automation of actuator control based on sensor readings, thus relieving the task associated with crop health and resource management.

*Figure 7 ESP32 microcontroller*

Integration of the ESP32 with Arduino brings flexibility into our project. In this respect, the ESP32 would be the main processing unit, while Arduino boards can be used to interface with specific sensors or actuators that require more pins or power. Such a setup allows for a modular structure in which the ESP32 takes care of communication and data processing, while Arduino boards handle physical interaction. Such collaboration eases the development process and enhances the comprehensive functionalities of the agricultural rover, allowing complex features like the integration of artificial intelligence models for predictive analytics and automation.

ESP8266

The ESP8266 is a small, low-cost Wi-Fi microcontroller that has recently gained much popularity for IoT applications due to its user-friendliness and advanced features. The ESP8266 makes it very convenient for developers to easily connect devices to the internet with an 80 MHz processor and a built-in Wi-Fi module. This technology can be applied in our project for data acquisition from various sensors and its transmission to a cloud platform or website in real time for monitoring. It will provide an appreciable rise in automation with regard to the irrigation systems and other agriculture activities that are controlled by actuators like pumps or motors through sensor data.



*Figure 8 ESP8266 microcontroller*

The addition of the ESP8266 brings greater capability to this project and eases development at several levels. In this setup, the ESP8266 is the main communication module, which handles Wi-Fi and allows data transfer, while the Arduino boards are used for auxiliary sensors and actuators that might require more GPIO pins or special interfacing capabilities. Integration allows the rover to have a modular and adaptive framework that allows it to acquire advanced capabilities in data acquisition and remote management while simultaneously allowing the integration of streamlined artificial intelligence models for enhancing real-time decision support.

Summary Table :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Arduino Mega | Raspberry Pi 4 Model B | ESP32 | ESP8266 |
| Advantage | Large number of I/O pins | Higher processing power (1.5  GHz) | Dual-core processor for higher  performance | Simple solution |
| High processing power | Multiple RAM  options (2GB, 4GB, 8GB) | Built-in Bluetooth support | Smaller size |
| Good for complex applications | Dual micro- HDMI ports for 4K output | More GPIO pins (typically 34) | Easier to use for basic applications |
| Supports multiple sensors/actuators | More USB ports (2  USB 3.0, 2  USB 2.0) | Supports more communication protocols (I2S,  CAN, etc.) | Good community support and  libraries |
| Greater number of I/O pins | Better low- power modes for battery applications | Sufficient performance for most IoT tasks |
| Disadvantage | Larger size may not fit all projects | Larger size may not fit all projects | Slightly more complex to set up | Only single- core processor |
|  | Higher power consumption | Higher power consumption | Higher power consumption | Limited GPIO pins |
|  | More complex to set up | More complex to set up for beginners | No Bluetooth supports |
|  |  | More expensive |  | Fewer advanced features |

Camera

Integration of the camera into our rover will significantly increase the functionality of the system with visual monitoring and data acquisition. Cameras can be deployed in agriculture for applications like crop health assessment, surveillance, or monitoring of the environment. The camera provides insights regarding the growing conditions and detects quite a few issues at an early stage, including ripeness and diseases. The selection of the camera is depending on the controllers we select and also on the resolution and quality it gives. We will discuss various camera options to select the one which enhance the rover’s functionally

Raspberry Pi Camera Module

The Raspberry Pi Camera Module is the best choice for our rover because it has better image quality and is perfectly compatible with Raspberry Pi boards. Resolutions of up to 12 megapixels and the ability to record video provide detailed visual representations of crops, hence facilitating effective monitoring and assessment of their health. Moreover, it works well with many models of Raspberry Pi; thus, its integration in our application should be easy.



*Figure 9 Raspberry Pi camera*

To communicate with the Raspberry Pi Camera Module, one can make use of the dedicated camera interface—CSI on the Raspberry Pi—which allows for easy configuration and programming using libraries like picamera.

ESP32 CAM:

The ESP32-CAM is an effective and cost-efficient option; it includes a camera module with an ESP32 microcontroller, making it a particularly suitable choice for our project. It supports Wi-Fi, which allows live video streaming or sending of pictures to a server to enable remote surveillance. It is an important feature, especially in the immediate analysis of data and informed decision-making within agricultural methodologies. Interfacing the ESP32-CAM with an Arduino is actually possible by programming the ESP32 using the Arduino IDE; it inherently has camera and Wi-Fi functions, so that was pretty easy to implement on our rover system.



*Figure 10 ESP32 Camera*

USB CAM

In our project we can use Arduino microcontroller to build a highly precise system for controlling the USB cameras which would allow for live data feed from cameras placed at different locations of the agricultural field. It will be possible to combine pairs of transmitter and receiver radios to develop a wireless configuration so as to enable the Arduino to switch on industrial or commercial cameras depending on the signal criteria. This flexibility allows proper configuration of the triggering methods; from grounding the trigger pin to applying voltage depending on the model of the camera.



*Figure 11 USB camera*

This system does not only help to find the best position for cameras in complex conditions but also increases data acquisition effectiveness for crop monitoring, 3D

measurements, remote sensing and other, which will certainly lead to agribusiness growth and our better analysis and decisions in this field. The figure shows the simple connection between Arduino board and camera

*Table 11 Cameras Comparison Table*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Camera Option** | **Resolution** | **Connectivity** | **Integration Ease** | **Advantages** | **Disadvantages** | **Price** |
| **Raspberry Pi Camera Module** | Up to 12 MP | CSI  (Raspberry Pi) | Easy with Raspberry Pi | High image quality, video recording | Only works with Raspberry Pi | 14.79$ |
| **ESP32- CAM** | 2 MP (up  to 5 MP) | Wi-Fi | Easy with Arduino | Small, affordable, and supports streaming | Lower resolution compared to others | 13.99$ |
| **USB CAM** | Varies (up to 1080p) | USB | Moderate | Flexible, easy to find, and offers good  quality. | Requires additional hardware (USB host  shield) to work | 18.5$ |

## Project Planning:

Embarking the project development and prototyping which aims to collect data soil and crop to assess crop ripeness and detect early stages of diseases minimizing the amount of losses and enhancing yield .The project is divided into 3 phases : Model Development , rover prototyping and website development .The planning phase serves as foundation for our development process by ensuring that we had enough knowledge about the constraints and issues that we may face and setting requirements and conducting feasibility study all in which help us in accomplishing out aim of this project . Therefore, we list the design and development specifications required for fulfilling our project aim:

* **Immediate Feedback**: The NPK sensor provides real-time data on soil nutrient levels, allowing farmers to make timely decisions regarding fertilizer application.
* **Precision Agriculture**: By understanding the exact nutrient composition of their soil, farmers can apply fertilizers more accurately, reducing waste and improving crop yields.
* **Remote Access**: Farmers can access real-time data about their crops from any location, enhancing their ability to manage their fields effectively.
* **Sustainability**: This objective aligns with sustainable agricultural practices by minimizing over-fertilization, which can lead to environmental degradation.
* **Scalability:** The technology developed can be scaled and adapted for various agricultural contexts, making it applicable to different crops and farming systems.

### Feasibility Study

A feasibility study is one of the major steps involved in assessing the possibility and viability of any project. It helps us identify the strengths and weaknesses of our proposed system, opportunities and threats present in the natural environment, resources needed to carry through, and finally the prospects of success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. In the case of agricultural robot project, the feasibility study addressed the following:

*Table 12 Feasibility Study*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Quantity | Price per unit | Total Price | Weight |
| Arduino Mega | 1 | 45$ | 45$ | 37g |
| Raspberry 4 Model B | 1 | 120$ | 120$ | 46g |
| Raspberry Pi Camera Module | 1 | 35$ | 35$ | 20g |
| Relay | 1 | 3$ | 3$ | 20g |
| MOTOR DRIVER | 1 | 5$ | 3$ | 50g |
| DC Motor | 4 | 15$ | 15$ | 400g |
| Rover Chassis | 1 | 50$ | 50$ | 1000g |
| Rs-485 | 1 | 8.99$ | 8.99$ | 10g |
| Spec 7 in 1 sensor | 1 | 48.11$ | 48.11$ | 50g |
| Total |  |  | 330.1$ | 1,633g |

The technical Feasibility involves the assessment of the availability and suitability of the required technologies like sensors, navigation systems, and software. This ensures that all conditions regarding sufficiency and availability of the required technologies for the functionality of the robot are attained within the guidelines of the projected scope. Estimating costs is one of the first steps to determine whether the project is financially feasible. It is involving the consideration of expenses concerning hardware acquisition development, and maintenance, along with unforeseen costs of upgrading to improve programming efficiency.

### Constraints:

1. Weather conditions: The design proposed in this paper operates ideally in clear weather conditions for multiple reasons. First, rainy weather may cause damage to the components used as they are mounted on top the rover, and it is not equipped to function in this type of atmosphere. Furthermore, the tomato crops are ideally grown in a sunny environment with high humidity, so growing the tomato plants in a rainy atmosphere would affect its growth.
2. Communication Range: A Wi-Fi Module is used to communicate soil data to the user through a website. The maximum range that can be covered by Wi-Fi module is 75 meters which means any distance beyond that range cannot receive any data.
3. Soil Quality and Navigation Accuracy: The design presented assumes the presence of flat soil level in the field to which the rover navigates accurately through its obstacle avoidance and navigation algorithms. However, if the soil was muddy/stony or not flat, the rover’s maneuvering through the field may be affected or limited.
4. Rover Weight**:** Soil in tomato fields tends to be loose, low density, and vulnerable to compaction. Small robotic rovers have high ground pressure due to low contact areas. Excess weight intensifies soil sinking, rutting and crop damage. By limiting total rover weight to a maximum of 50 kg and distributing across multiple wheels (4 in this case) with sufficiently wide terrain-treading tires, recommended ground pressure for agricultural soils can be achieved: - around 0.7 psi (~48 kPa). This threshold ground pressure prevents significant soil compaction while posing minimal risk of damage to developing tomato plants during farm operations.
5. Rover Speed**:** To ensure precise dispensation of seeds across the tomato field during the planting phase, the maximum speed of the rover is limited to 1 m/s. Exceeding this speed threshold while attempting to calculate and coordinate seed drop locations can result in missed spots within the field area. Our implementation shows speed readings of around 0.5 m/s with a PID accuracy of 90%, meaning that there is a small offset that needs to be assigned whenever there is a new target speed based on the current conditions of the terrain.

### Standards:

Environmental ISO (International Standards Organization):

1. **ISO 17989**, Tractors, and machinery for agriculture – Sustainability. *This standard gives manufacturers of tractors and agricultural machinery the guidance they need to integrate sustainability principles into the whole life cycle of their products. The standard applies specifically, to equipment used in food, fibers, fuel and lumber production for humans and livestock.*
2. **ISO/TC 23**, Tractors and machinery for agriculture and forestry. *Oversees various areas like safety, testing, crop protection gear, operator controls, forestry machines, irrigation tools, drainage equipment, and electronics through its subcommittees.*

**ISO 4002**, Equipment for sowing and planting (lies under ISO/TC 23)

1. **ISO/TC 134**, Fertilizers, and soil conditioners. *ISO has a range of standards for measuring the content of fertilizers and assisting with the sampling process. These include standards for measuring levels of chemicals in fertilizers such as nitrogen, ammonium nitrate, phosphorus, and potassium, as well as standards defining the basic terminology, sampling methods and test procedures for determining the bulk density of diverse types of fertilizer.*
2. **ISO 15003**, Electrical and electronic equipment. *sets rules and advice for makers of electronic gear in farm machinery. It outlines tests for tough conditions faced during agricultural work, like extreme weather or rough terrain.*
3. **ISO 18400-104**:2018, Soil Quality Standards that cover soil sampling strategies and methods, ensuring consistency in soil condition measurements. *This document gives general guides on the average property of soil, the variability of soil, and the spatial distribution of a variety of properties.*
4. **ISO 10218-1:2011**, Robots and robotic devices — *Safety requirements for industrial robots specifies requirements and guidelines for the inherent safe design, protective measures and information for use of industrial robots. It describes basic hazards associated with robots and provides requirements to eliminate, or adequately reduce, the risks associated with these hazards.*

*Table 13: Used Standards for the Agricultural Rover*

|  |  |
| --- | --- |
| *Standards* | *description* |
| **ISO 17989** | Tractors, and machinery for agriculture – Sustainability |
| **ISO/TC 23** | Tractors and machinery for agriculture and forestry |
| **ISO/TC 134** | Fertilizers, and soil conditioners |
| **ISO 15003** | Electrical and electronic equipment |
| **ISO 18400-104:2018** | Soil Quality Standards |
| **ISO 10218-1:2011** | Robots and robotic devices safety |

### Team Members Tasks

* + Project Manager: Nour Hijazi
  + AI Developer: Nour Hijazi
  + IoT Developers: Nour, Reem, Souha, Zeina
  + Web Developer: Souha, Zeina, Reem

### Software Model :

In the initial stage of our project, we follow the Waterfall methodology to ensure that the system will meet the farmer's requirements. Requirements need to be clearly drawn and should be focused on how to monitor soil conditions, control the rover, and incorporate AI for crop assessments. In addition, thorough documentation develops a detailed document representing all functional and non-functional requirements, and use cases including admin and user side.

In the development phase, we shift to the Scrum methodology that facilitates agile project management. The project is divided into sprints for the development of specific features and functionalities such as real-time data visualization and control of the rover remotely. Scrum meetings are held daily to discuss the progress of the work, sort out the challenges if any, and keep the team on the same page regarding the goal of the project. In an environment like this, adaptation to changes is so easy and quick because the focus is on the delivery of value.

By integrating the structured clarity of the Waterfall methodology with the iterative nature of Scrum, we develop a really robust framework with which to approach the project of the agricultural robot. The early phases give one a very solid grounding of well-defined requirements, while the Scrum model takes care of continuous feedback during development. This will ensure incremental feature delivery that actually meets user needs in the real world, and has kept the project responsive at every input from stakeholders, culminating in enhancing the practice of agriculture.

### Project Issues :

The development of the proposed project that offers a rover that is used for soil and crop assessment and monitoring present wide range of project issues that must be concerned and if its possible address them to ensure that the project will end with successful implementation .The project issues could impact the projects’ reliability and effectiveness some of the main issues are presented below as follows :

### Technical Issues:

**Sensor accuracy :** The soil sensor utilized in this project is NPK which measures nutrients in soil ensuring the sensors accurate reading is crucial such that inaccurate sensor reading could lead to improper fertilizer usage which impact the crop yield and health .To resolve this issue we should frequently calibrate the sensor for accurate reading and treatment

**Model performance :** The models used serves two main goals detect diseased crop based on their leaved and classify ripe and unripe crop. The model proposed could have high evaluation metrics yet fail at classification and detection due to bad quality images captured by the rover ‘s camera since model’s performance could be altered by cluttered background ,image illumination , angle , and distance affecting the image quality. To resolve this issue, we could use high resolution camera and perform data augmentation .

### Operational Issues:

**Rover’s navigation :**The rover must be able to move in uneven surfaces and different field layouts and we may face issue while using gps that may provide inaccurate positioning because of signal loss and jamming that we are facing in Lebanon due to Israeli conflict.

**Battery life and operation duration** : the rover operation duration depends on the battery capacity and the power consumed by the sensors and motors used such that the limited battery life restricts or reduce the operational time of the rover.

### Tools /Technology:

Python:

Python has dynamic semantics which is a high-level, object-oriented, interpreted programming language. It is highly appealing for Rapid Application Development for use as a scripting or glue language to join pre-existing components because of its high- level built-in data structures, dynamic typing, and dynamic binding.

PyTorch

The Python programming language and the Torch library serve as the foundation for the open source PyTorch machine learning (ML) system. Torch is an open-source machine learning library for building deep neural networks.

Javascript

JavaScript is a scripting programming language that allows you to generate dynamically updated information, handle multimedia, animate graphics.

AI Model:

Pycharm : is a Python-specific Integrated Development Environment (IDE) that includes a variety of critical tools for Python developers. These tools are tightly integrated to offer a pleasant environment for productive Python, web, AI and data science development.

Google Colab :is a hosted Jupyter Notebook service that requires no setup and offers free computing resources such as GPUs and TPUs.Colab is particularly well-suited for machine learning, data science, and teaching.

Ultralytics : is the home of cutting-edge computer vision models used for picture classification, object identification, image segmentation, and posture estimation. Ultralytics also provide us with pre-built procedures for training, fine-tuning, and applying these models via an easy-to-use API.

1. Concerning Dataset

Kaggle : is a platform for data science and machine learning specialists where they may compete to build the best models for addressing specific issues, analyzing specific data or search for datasets.

Roboflow: enables developers to create their own computer vision apps, regardless of their proficiency or experience level , and showcase some open-source dataset.

Google Dataset Search : is a metadata search engine that searches for millions of datasets across thousands of sources on the web. Google Dataset Search allows us to find datasets anywhere they are housed, whether on a publisher's website, a digital library, or an author's personal webpage.

Web Development:

Frontend:

React JS: is a frontend-focused JS library for web user interface. It’s creates user interfaces by combining separate JavaScript components.

Backend:

Flask: is a lightweight and adaptable framework for web development and web applications that uses the Python programming language. Flask, with its minimalistic design and feature-rich features, is a popular choice among development teams for creating dynamic webpages, APIs, and microservices.

Mongo DB :is an open source, NoSQL database that uses flexible documents instead of tables and rows to process and store various forms of data

MQTT (Message Queuing Telemetry Transport) is an IoT real time messaging protocol used by devices for communication. It designed to perform well under low bandwidth, high latency or unreliable networks. It uses publisher–subscriber model of messaging where one or more publishers can send messages to a broker that forwarding them to subscribers.

WebSocket is aa communication protocol mainly about how a client and a server will be able to share content through the full-duplex pipe. That is, two directions of data interchange, where the client and the server are also participating, exchange information with and within one another in turns over time.

AWS

Amazon Web Service, or AWS, is an online platform that offers cost-effective, scalable cloud computing solutions. It provides a variety of on-demand services to assist corporations and organizations in growing, including computational power, content distribution, database storage, and more.

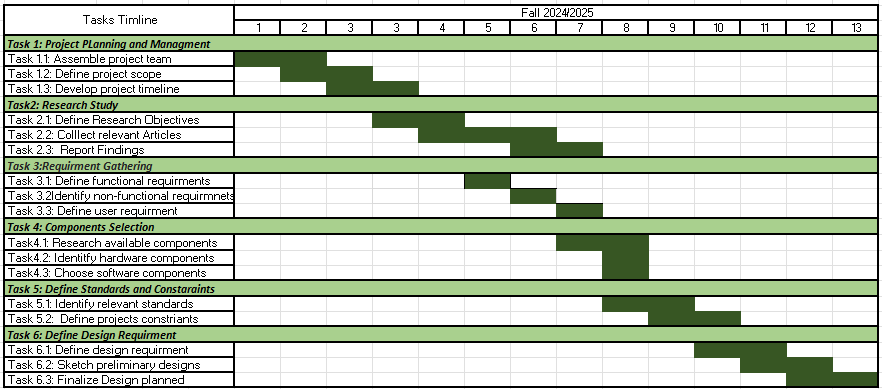
### Milestones:

* 1. Project Planning and Management:
* Setting Objectives: We define the overall goals and objectives of the project- increasing agricultural yield and sustainability, enhancement of real-time insights to farmers. Additionally, define what we are serving and who our target users.
* Deliverables: Clearly describe what specific outputs are expected at various levels, such as a functional prototype of the rover, AI models, and a user-friendly website.
* Timeline Development: Outline the development timeline of the project using

various project management tools and the development of a Gantt chart. Case use on the setting of a website: state the privileges that exist between a user and the admins. The user of our project is the farmer or landowner who has privileges to access the homepage that will show data acquired to outline phases-1, 2, and 3, tasks, and deadlines. This will be beneficial in ensuring that the project stays on track and the group members are well conversant with their tasks.

* 1. Study and Research of Feasibility:
* Research on Technologies: Deeply research the existing technologies of soil sensors, cameras, and AI models and tools used to accomplish all phases. This will be a study of the current solutions in the market and finding the gap that your project could fill.
* Alternative Design Analysis: Review alternative design approaches for each phase; where our project we divided the project into 3 phases: (Phase 1 - AI Model Development, Phase 2: Rover Design and Prototyping ,Phase 3 : Website Development )
  1. Requirements Gathering
* Where we set functional requirements for the defined phases-for example, Rover: Mobility; for AI Models: Crop Detection Accuracy
  1. Selection of Component:
* Hardware Components: Researching and selecting appropriate sensors, including RGB cameras for image capture and soil moisture sensors.
* Software Tools: Chose the programming languages according to project needs- for example, for AI models in Python due to a vast number of its libraries ; and JavaScript for implementing the interactive parts of the website.
  1. Initial Design and Prototyping:
* System Architecture Design: Describe how the various components of the system will interact. This means explaining how data will be captured from sensors, into AI models, and down to the website, where the user can access insights with some kind of standard in place. • Data Preprocessing: This involves collecting datasets relevant for training AI models. This might involve gathering images of healthy and diseased crops and then annotating them.
  1. Identify the Needs of the Project and the Potential Risks:
     + Risk Analysis: Carry out the risk analysis, pinpointing potential issues that may arise with hardware failures-sensor malfunction or model prediction inaccuracies, for example, false positives in disease detection.

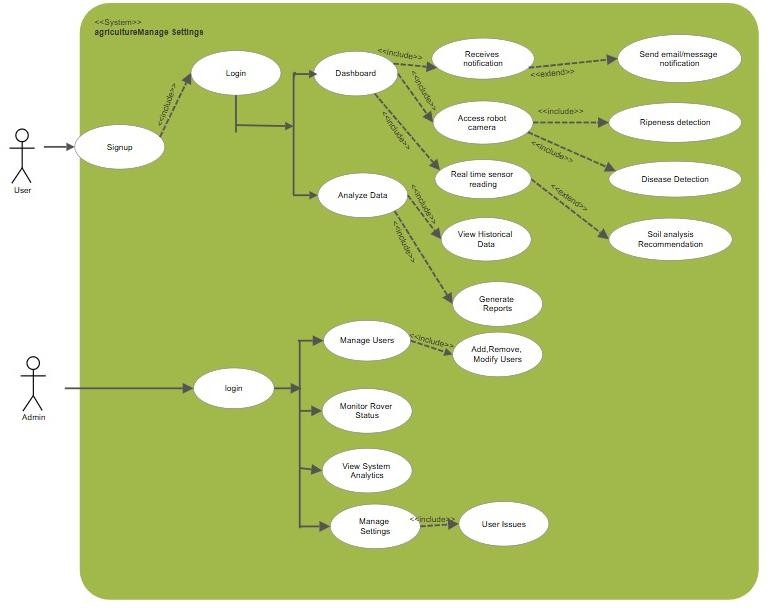
### Summary Grantt chart :

****

## Requirements

One of the crucial aspects of any successful project is the careful identification and specification of requirements that must be met upon completion of this project . This chapter focuses on establishing the necessary user, system, functional, and non- functional requirements for our project, which involves developing a rover equipped with sensors to monitor environmental conditions and provide real-time insights into crop health. By clearly defining these requirements, we aim to ensure that the final product effectively addresses the needs of its users, including farmers and landowners.

### Use Case :

****

*Figure 43 Use Case Diagram*

#### User Requirements

* User-Friendly Interface: The interface should be intuitive and user-friendly; the user should be able to work his or her way through the application easily across all functions and features. It needs to be responsive because it will run on smartphones, tablets or laptops.
* Realtime Notifications: It should notify the user in real time in the case of crop diseases and deficiencies detection through alerts, which come into the system alongside the detection of the soil condition through different means. The same should provide notification to a user that he can customize based upon his requirement, selecting accordingly where to deliver the same using e-mail or SMS mode.
* Easy Accessibility of Historical Data: This solution should allow easy accessibility of historical data to the end user for trend analysis, decision-making, filtering of data, and its exportation for review purposes.
* Regular Reports: The system should prepare and send such reports, which summarize findings and recommendations, automatically on a daily or weekly basis. It should provide options of content and form as far as possible with which the user can tailor the reports.
* Support and Feedback Mechanism: It should be able to let the user report a problem or request a feature to an integrated feedback mechanism which responds immediately upon receipt, and in case of taking action later.
* Monitoring and Control: Provide the user with the ability to track, in real time, the status and location of the rover, including operational metrics such as remaining battery life. It should also be able to perform remote operations on the rover if such a need arises.

#### System Requirements:

* Hardware Requirements: The robot should connect to high-quality sensors that will measure the moisture and nutrient levels of the soil, while cameras will do crop analysis. It has to be connected with a reliable microcontroller to handle data collection and the communication between its parts.
* Database Management: It therefore requires the use of strong databases for safe storage of sensor readings, user information, and historical data of analytics. Secondly, periodic back-up implementations will not allow any sort of data losses in this concern.
* Backend Infrastructure: The system needs to be deployed on a fully-fledged backend server to handle every request and process data in real time without any latency. In addition, scaling could be done in which more loads of data would be handled, for which cloud integrations may be considered.
* Communication Protocols: The data transmission between sensors, microcontrollers, and the backend server should be very stable, using reliable

communication protocols. Error detection and handling mechanisms are highly necessary.

* Scalability: It requires a modular system architecture to be adopted that will provide a system where other sensors or additional users are easily integrated with negligible rework. The ability in support of heightened demand should also be assessed with mechanisms of performance monitoring included.
* Security Measures: Data encryption has to be in place to secure sensitive data while moving and data at rest. Strong owner account authentication should be provided. It could be two-factor authentication.

#### Functional Requirements:

* + Data Collection: The system should be able to provide the facility of data collection from integrated sensors for soil, crop, and leaf analysis. All the data collected should be logged with a timestamp for future reference. The collected data should be logged with a timestamp for further use
  + User Notifications: The System has to be capable of triggering alerts automatically while crossing some threshold, low moisture amongst other factors. Users should be allowed to change notification preference wherever possible.
  + Data Storage: The outputs from the sensors and analyses should be securely saved in a database so that the user can have easy access to these data. It will therefore be of paramount importance that data retention policies be implemented with respect to data stored regarding how long it has to be maintained.
  + Historical Data Access: The user interface should be provided to users for easy navigation of historical data. Basic analytics should be available to visualize the trend and correlation in the data.
  + Sensor Communication: Continuous communication shall be ensured between sensors and the microcontroller to reduce data loss. Real-time status checking of sensor operations by users will be permitted.
  + Data Handling Capability: The system should efficiently manage processing large volumes from many sensors without a degraded performance. The methodologies of handling this data should have compressive natures to optimize storage and transmission.
  + Machine Learning Models: It shall adopt machine learning algorithms that will determine the ripeness of the crops by analyzing images of the crops. It shall detect leaf diseases using an advanced image processing technique.
  + Backend Server Implementation: The backend server should be one that provides a well-documented API through which the interactions between the frontend and the backend take place with absolutely no hassle. There has to be mechanisms put in place that ensure the integrity of data during interactions.
  + Emergency Alerts: The system should be designed to monitor critical conditions that may arise and send alerts at once to users when these situations do arise. All alerts should contain recommendations in regard to the best action against such a situation.
  + Support System: Users should have one channel for issues and feature requests through an integrated support system that allows tracking of query statuses.
  + Dashboard Features: It should provide real-time status updates of the rover, including camera feeds and sensor readings, on the dashboard. The status of the various parameters should be represented using visual indicators, which can easily pinpoint problems.
  + Real-time Communication: For the same reason, low-latency data processing of this system is also necessary so that immediate analysis and decision-making could be achieved. Give responses right away based on the action performed by the users on this system. Various parameters are supposed to have their status indicated using visual indicators, showing problems easily.
  + Real-Time Processing: It is also necessary that the system should avail data processing with low latency to enable timely analysis and decision-making. Immediately show feedback for whatever action that the user commits within the system.

#### Non-Functional Requirements:

* + - Performance and Reliability of Operation: The system performance shall be at 99.9% uptime, at which the users can use without any disruption. This degree of reliability will require intense infrastructure together with effective fail-over mechanisms against potential disruptions.
    - Easy to Update and Maintain: The system is such that it will be made really easy for updates whenever its maintenance does so; thereby allowing for quicker fixing of bugs and feature enhancements without actually taking it out for a very long time, the architecture would lend to a quick deployment in respect of updates. Modifications could easily be made based on the fact that any application can have well-written documents.
    - Data Security and User Authentication: This will be utilized in the protection of user data and system communications through encryption, among other ways that bar access to unauthorized people. Stronger forms of user authentication shall be employed for verification, such as multi-factor authentication techniques.
    - Intuitive and User-Friendly Interface: The user interface should make sense and be accessible to users at every level of technical proficiency. It will also include clear labeling, guided tutorials, and accessibility features that make it even easier for all users.
    - Aesthetically Pleasing Design: If the website is visually attractive-beautiful images, modern layout, and modern agricultural contexts among others-then it represents an interesting interface for use improvement and regular interactions.
    - Environmental Adaptability: It should work accordingly in every condition, right from the environment to the temperature and humidity ranges. This will call for durable hardware and software solutions in potentially harsh agricultural environments.
    - Minimum Energy Consumption: The system should consume little energy to minimize operation cost and environmental impact, including the use of energy-saving components and power-saving mode in case of no action.
    - Minimization of Operational Cost: The system should minimize the cost of operations by simplifying procedures and reducing human interventions. It would, therefore, depend on automation and management of resources in an effective manner.

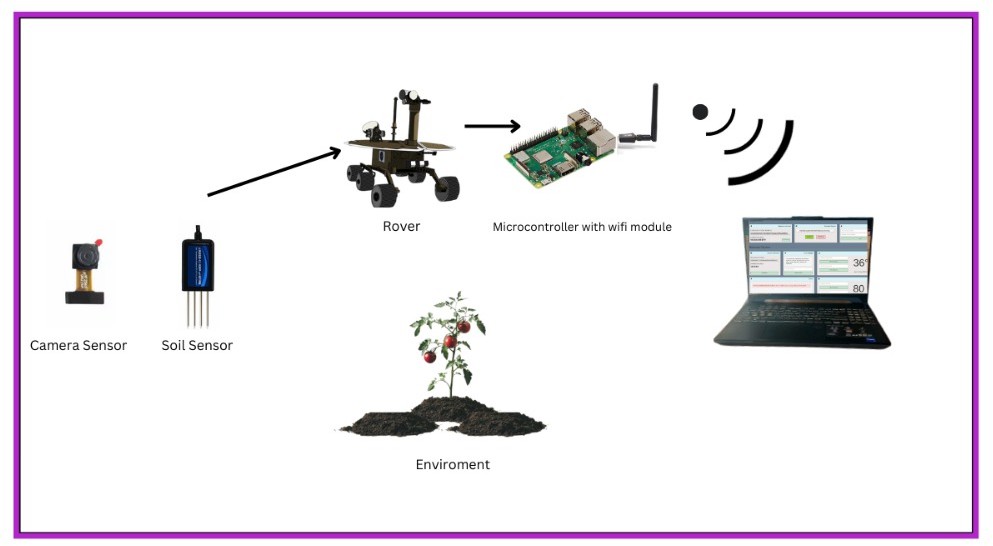
## Project Planning:

Embarking the project development and prototyping which aims to collect data soil and crop to assess crop ripeness and detect early stages of diseases minimizing the amount of losses and enhancing yield .The project is divided into 3 phases : Model Development , rover prototyping and website development .The planning phase serves as foundation for our development process by ensuring that we had enough knowledge about the constraints and issues that we may face and setting requirements and conducting feasibility study all in which help us in accomplishing out aim of this project . Therefore, we list the design and development specifications required for fulfilling our project aim:

* **Immediate Feedback**: The NPK sensor provides real-time data on soil nutrient levels, allowing farmers to make timely decisions regarding fertilizer application.
* **Precision Agriculture**: By understanding the exact nutrient composition of their soil, farmers can apply fertilizers more accurately, reducing waste and improving crop yields.
* **Remote Access**: Farmers can access real-time data about their crops from any location, enhancing their ability to manage their fields effectively.
* **Sustainability**: This objective aligns with sustainable agricultural practices by minimizing over-fertilization, which can lead to environmental degradation.
* **Scalability:** The technology developed can be scaled and adapted for various agricultural contexts, making it applicable to different crops and farming systems.

### Conceptual Design

Formulating a conceptual Design is important step in project planning offering some overview of the project methodology and tools used to achieve each step Figure 2 below presents the proposed project conceptual design .



*Figure 12 Conceptual Design of the proposed project*

### Project Organization:

The project proposed offers a rover that is equipped with a camera and soil sensor and 2 AI models that classifies crops as ripe and unripe and classifies diseases of crops based on their leaves. To lower time required, were most of agriculture tasks are labor- intensive. Starting with the first step that is data acquisition to assess the crop health, we need to look up 2 major things: the soil of the plant and the crop itself. Regarding soil assessment, we choose the use of an NPK soil sensor that measures the soil nutrients to determine the exact amount of soil nutrients without going through conventional time- consuming processes, and it provides instant insight about the soil's nutrient levels, allowing for accurate prediction of nutrients needed by fertilizers further details about the sensor will be provided in phase 2 of the project .

Now concerning the plant health assessment, we choose to equip the rover with a camera that is used to monitor the crop offering landowners remote access to their land whenever they are, wherever they are. This camera is also used to capture images or video streams of the crop present so they can be later on assessed using an AI model. In our project, we choose to assess the crop health using 2 AI models, each offering different services, which will be explained in details in phase 1 of the project . Crops health could be assessed using different aspects but mainly the crop's health is determined assessing` it's look and its leaves, such that leaves can show early signs of the crops disease which can help us in aiding it with the medications needed so we could limit or prevent losses. However, checking each and every crop leaves manually is extremely time- and labor consuming also it requires expertise to detect the true disease the plant is suffering for further treatment. For precise assessment, our model not only detects diseased crops but also provides disease names, which aids farmers in precise disease treatment Leaves health [[15] ,[16].](#_bookmark74)

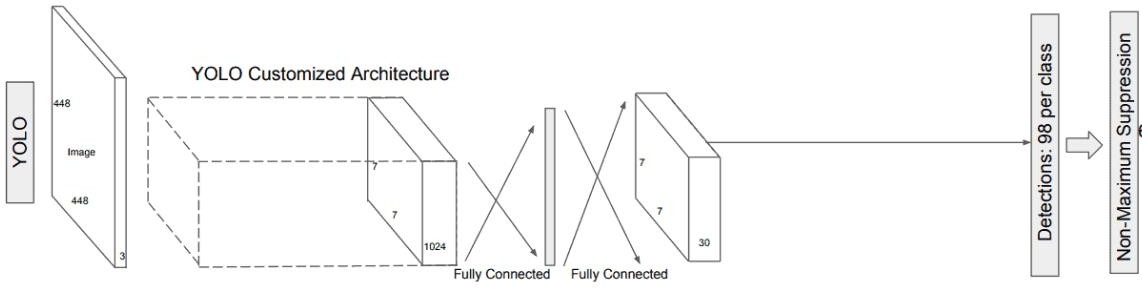
Furthermore , we’ll utilized an AI model that classifies the crops as ripe and unripe, which provides a slight insight for farmers to when to harvest there lands precisely .In addition to AI model that detects crop diseases based on their leaves which help farmers increase their yield by treating diseased crops. Secondly after data acquisition the data taken from sensors that on the rover through wired connection ,the data captured are then transmitted through a microcontrollers that have built in Wi-Fi module that transmit them wirelessly to be interfaced through a website. The website offers some user- friendly interface taking into consideration that most farmers and landowner are not familiar with such technology ,showcasing data captured visual representations such as histogram, bar graphs to easing the accessibility of the website .

### Phase 1 :AI Model Development

In this project we are proposing two AI models each has its own objective .The model should be able to accurately classify ripe and unripe crop in addition to detect diseased plants based on their leaves providing the diseased detected .Prior choosing the optimal object detection model we are going to view some alternative design to ensure the model chosen it the optimum .

#### Optimal Model Selection :

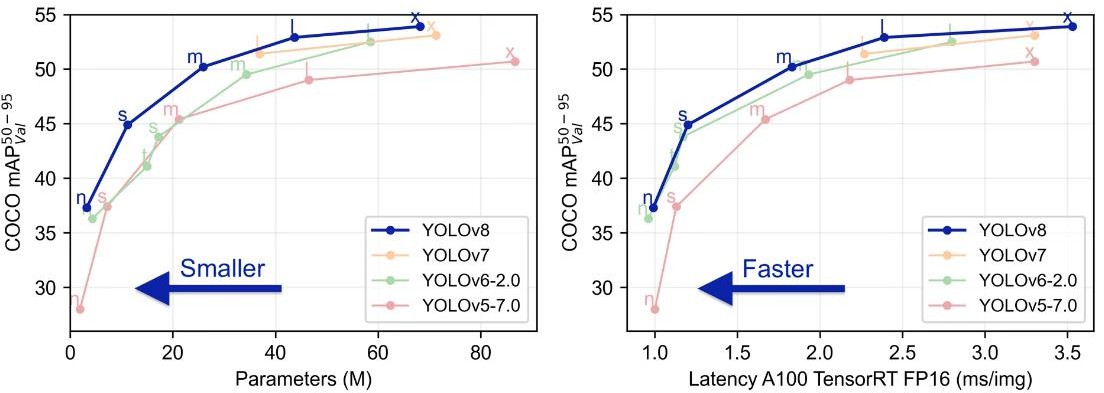
After thoroughly reviewing the latest literature and the some of the alternative design regarding agricultural applications where environment cluttered with branches and trees, it is realized that many prominent object detection models, despite their high accuracy in controlled settings with unique specifications and characteristics as shown in RetinaNet model and Mask R-CNN they struggle to effectively detect such complex conditions they failed to meet the results got from YOLO Model . YOLO (You Only Look Once) model has consistently demonstrated superior performance outperforming all previously mentioned models in challenging environments. The architecture of YOLO meets our project aims for real-time applications, offering low inference times and high frames per second (FPS) while maintaining high accuracy levels. This makes YOLO the optimal choice for our project, which aims to enhance crop monitoring and management through efficient and reliable detection systems.



*Figure 13 YOLO Architecture[19]*

Yet YOLO have different versions each having its own feature characteristics . it’s notable to mention that we will not compare YOLO version before YOLOv5 since they are considered as “out dated” and their inefficiency in detecting objects inspired the release of newer versions .

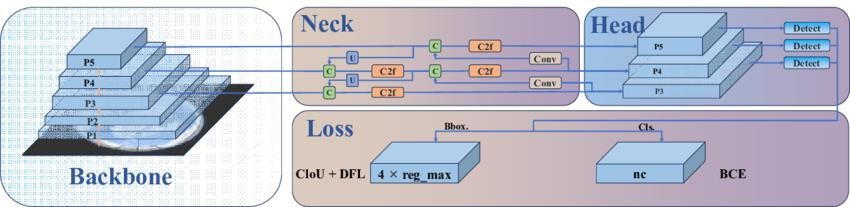
To accurately choose the most optimum model we will compare the YOLO models considering accuracy ( presented as mAP50-95%) with respect to parameters and the accuracy with respect to latency( ms/img) as shown in figure below :



*Figure 14 YOLO models comparative analysis[26]*

The following graph compares the performance of four YOLO models (YOLOv5 which is presented in pink color,YOLOv6 which is presented in green color , YOLOv7 which is presented in orange color , and YOLOv8 which is presented in blue color ) .As shown in graph that compares accuracy with respect to parameters it shows that YOLOv5 model requires the least parameters yet it sacrificed the some of the accuracy reaching approximately 51 underperforming all YOLO models present in this comparison , while this graph showed that YOLOv8 outperformed all other models reaching value of 54 .On the other hand , it was shown in the graph that compares accuracy with respect to latency that YOLOv6 is slightly the fastest yet it scarified some of its accuracy not reaching the highest accuracy with respect to other models .On the other hand YOLOv5 and YOLOv8 showed similar low latencies yet YOLOv8 achieved better accuracy compared to all other models present in this study which implies that YOLOv8 is the optimum model for our project .

The optimum model selected is divided into 4 major parts (Backbone ,Neck, Head and Loss) YOLOv8 has its custom backbone architecture, CSPDarknet53, using cross-stage partial connections for easier feature propagation and interaction between layers. The backbone forms an essential component in the extraction of informative features from input images; it captures simple patterns like edges and textures in the early layers while providing a hierarchical representation as shown in the figure below :



*Figure 15 YOLOv8 architecture, divided into four parts: Backbone, Neck, Head and Loss[27]*

that allows for extracting features at multiple scales. While at the backbone of it, CSPDarknet53 optimizes for speed and accuracy, enabling itself to grasp low-level textures and high-level semantic information that has to be captured for real object detection.

The neck serves as a bridge between the backbone and head, where all feature fusion operations are performed. As for YOLOv8, instead of classic Feature Pyramid Networks, it uses a new type of module called C2f. This module merges the feature maps obtained from the different stages of the backbone in a very effective manner, ensuring that the network can detect objects of varied sizes. It achieves this balance by using contextual information to increase its detection accuracy without significantly increasing computation through reduced spatial resolution. The architecture of the neck makes it suitable for object detection tasks involving small objects because of its ability to integrate high-level semantic features with low-level spatial information.

The head is the final part of the YOVOv8 architecture, responsible for the generation of outputs, bounding boxes, confidence scores, and class labels of the detected objects. This model abandons the anchor-based methods used by its predecessors for bounding box prediction in favor of anchor-free approaches.

Ihe approach makes a direct prediction of the mid-point of an object instead of the anchor boxes. It generates bounding boxes associated with the potential objects in an image, assigns confidence scores with regard to the likeliness of the presence of an object, and sorts the detected objects according to their respective categories.

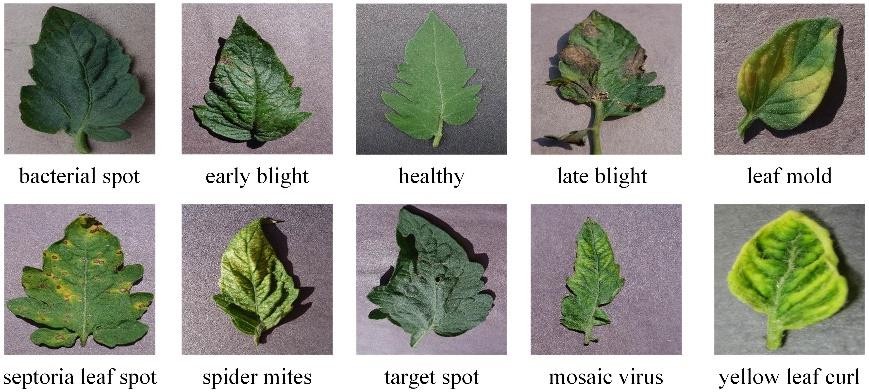
YOLOv8 uses a modified loss function that contains several components for optimal training. It includes terms of bounding box regression, objectness score prediction, and classification loss. The design of the loss function helps in improving convergence during training while ensuring that the model effectively learns to distinguish between classes

### Model 1 : Disease Detection

One of the methods to asses crop health is determine whether the crop is diseased or not in this section we are going to utilize YOLOv8 model to detect crop disease based on their leaves which detect diseases in early stages which aids farmers and landowners in treating them limiting the losses and increasing yield.

#### Data Collection :

The dataset used in this project serves as an integral part of the analysis and building of the model. It was obtained from a publicly available source, namely Roboflow, an open repository known for its quality and relevance to the research topic. The dataset being used here has 4,128 images of leaves having different types of diseases while some images contain leaves that are healthy

.

*Figure 16 Dataset classes*

The Dataset is divided into 9 different classes ,1 of the classes is for healthy crop and 8 of which are related to unhealthy diseased crop leaves that are named as the following :

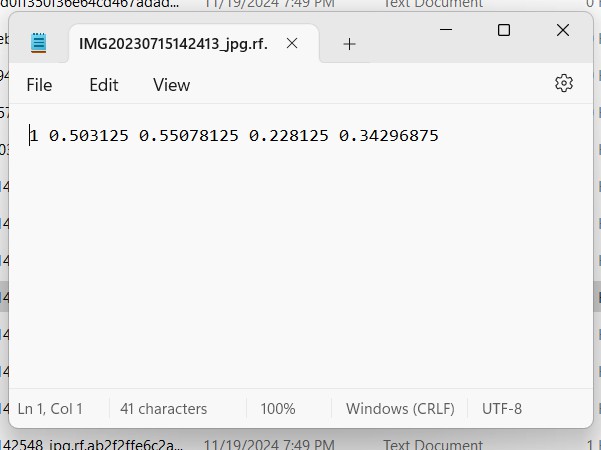
* Healthy
* Early Blight
* Late Blight
* Bacteria spot
* Leaf Mold
* Mosaic Virus
* Septoria leaf spot
* Spider Mites
* Yellow Leaf Curl Virus

The input files will be given in two formats :

1. The raw image.
2. The annotated raw image in Yolo format (specifically YOLOv8)



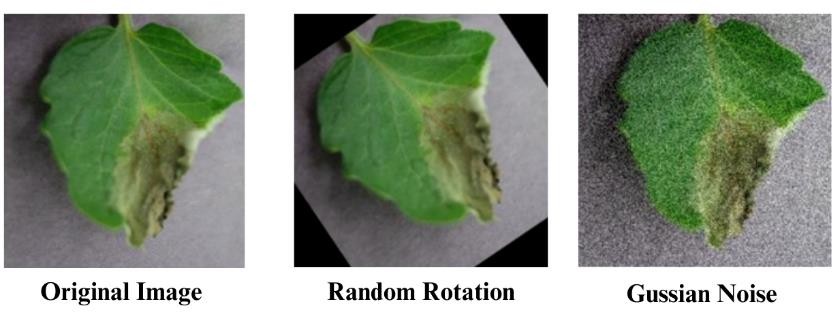
*Figure 17 Raw image*

**

*Figure 18 Annotated raw image in YOLOv8 Format*

#### Data Preprocessing :

Once the data is collected disease crop images is completed, the first step is screening and cropping. Images with blurred and large light spots in the diseased portion are eliminated, and images with a moderate size of disease are cropped. The images are adjusted to have resolution to 640 × 640 in the second step, followed by data enhancement. The enhancement methods include image rotation, brightness adjustment, and . Image rotation and scaling simulate various angles and distances during detection, while adjusting brightness and adding Gaussian noise mimic different external lighting conditions. These data augmentation techniques effectively simulate diverse external environments .The data augmentation methods simulate various external environments, enhancing the diversity and complexity of the dataset. This approach improves the usability and robustness of the disease detection model. The improved picture data is displayed in Figure 13 below:



*Figure 19 Data Preprocessing leaf Disease Dataset*

The dataset was separated as follows 80% for training , 10% for testing, 10% for validation . These ratios ensures that the model receives sufficient samples for learning, his method is utilized due to the fact that training the data one dataset makes it bias to the images present in the dataset where the model will be able to classify diseased or healthy leaves efficiently with the images present in the dataset , and poorly classify new images .This approach prevents the model from overfitting , by dividing the data into subsets we can accurately assesses how well the model generalizes on unseen images . allowing it to adequately capture the disease features in the dataset.

### Model 2 : Ripeness Detection

AI model for the identification of crop, whether ripe or unripe, is important in investigating crop health, as it would afford farmers real-time analyses of the maturity stages for their produce. Precise identification of fruit ripeness enables better harvesting scheduling that reduces waste and enhances total yield quality, generally contributing to more viable or sustainable of agriculture practices .In this section we are going to utilize YOLOv8 model for detecting ripe and unripe crop based on specific dataset

#### Data Collection :

The dataset used in this project serves as an integral part of the analysis and building of the model. It was obtained from a publicly available source, called Roboflow which is an open repository known for its quality and relevance to the research topic. The dataset being used here has 2607 images of ripe and immature tomatoes having different shape

,sizes and color.



*Figure 20 classes for tomato ripeness Dataset*

The Dataset is divided into 2 different classes ,1 of the classes corresponds for ripe tomatoes and the other class belong to immature tomatoes that are named as the following :

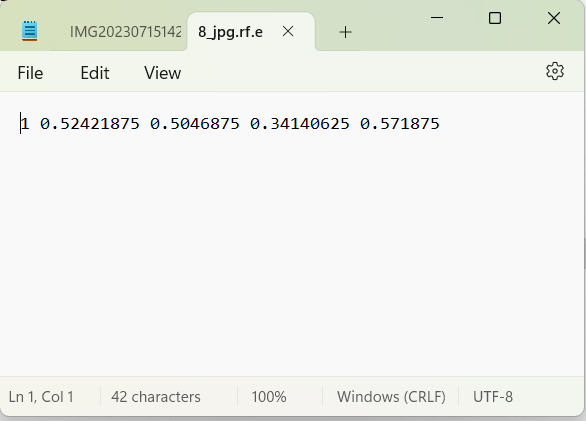
* ripe
* unripe

The input files will be given in two formats :

1. The raw image.
2. The annotated raw image in Yolo format (specifically YOLOv8)



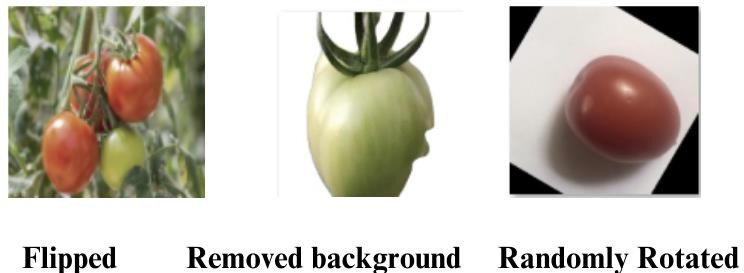
*Figure 21 Raw image for tomato dataset*

**

*Figure 22 Annotated raw image for tomato dataset in YOLOv8 Format*

#### Data Preprocessing :

Once the data is collected (ripe and unripe crop images) is complete, the first step is screening and cropping. Images with blurred and large light spots in the ripe portion are eliminated, and images with a moderate unripe crop size are cropped. The images are adjusted to have a resolution of 640 × 640 in the second step, followed by data enhancement. The enhancement methods include image rotation, brightness adjustment, and. Image rotation and scaling simulate various angles and distances during detection while adjusting brightness and adding Gaussian noise to mimic different external lighting conditions as shown below where we flipped some of the images, randomly rotated the images, and removed the background of some images and this could help generalize the model since due to similar background the model could consider that the white background is a feature and get confused when tested with new images having different backgrounds. These data augmentation techniques effectively simulate diverse external environments. The data augmentation methods simulate various external environments, enhancing the diversity and complexity of the dataset. This approach improves the usability and robustness of the disease detection model. The improved picture data is displayed in Figure 12 below:



*Figure 23 Data Preprocessing tomato Dataset*

The dataset is also separated here as follows 80% for training, 10% for testing, and 10% for validation to ensure the model generalizes on unseen images

#### Training Model :

In preparation for model training, the dataset was organized and divided into subsets, as described previously, which included thousands of labeled images. The dataset was structured to align with the requirements of YOLO training, featuring image files alongside corresponding label files in a YAML file. The dataset was divided into different subsets for 80 % training, 10 % validation, and 10% testing, as mentioned in the data preprocessing section above. The YOLOv8 model was rigorously calibrated with appropriate hyperparameters, including learning rate, batch size, and the number of training epochs, all aimed at enhancing its performance.

## Phase 2 :Rover Design and Prototyping

### Hardware Components :

* 1. Sensors a- SPEC (7 in 1 sensor) The SPEC sensor presented in Figure 3 integrates soil moisture, NPK, and pH sensors into a single unit. Soil moisture measurement is utilized to assess the volumetric water content in the soil. An electromagnetic wave transmitted between two exposed pads gauges the duration it takes for the wave to traverse from one rod to the other, providing insights into the soil's ability to store electric charge and its volumetric water content. Additionally, soil fertility is determined by measuring the levels of Nitrogen (N), Phosphorus (P), and Potassium

(K) ions in the field this product navigates. Ion-selective electrodes, designed for specific ion detection, generate electrical signals proportionate to the concentration of each ion upon contact with the soil. This measurement is crucial, as these nutrients play a vital role in crop growth in an agricultural setting.



*Figure 24 SPEC 7 in 1 Sensor*

Furthermore, for pH measurement, when the two electrodes of the soil pH sensor are inserted into the soil, they utilize voltage changes associated with ion-sensitive field- effect transistors to measure soil pH levels. Soil pH significantly influences plant growth by affecting nutrient availability and solubility in soil water, with optimal

plant development occurring in soil with a neutral pH (6-7). The data provided by the SPEC sensor is crucial for the design as it is used to generate a grid on a web application that displays the moisture levels, NPK readings and the PH value across different coordinates within the field. This gives insights to users into the field conditions they are dealing with, helping them stay informed and ready to take any necessary actions when needed. The power supply for this sensor system operates within the range of 12-24V DC. For moisture levels, the sensor offers a precise measurement range from 0 to 100%RH. The pH measuring range is between 3 and 9 pH. NPK measurement spans from 0 to 1999 mg/kg, providing insights into crucial nutrient concentrations for optimal plant growth. The electrical conductivity (EC) measuring range extends up to 20,000us/cm. The sensor has a moisture precision at

±2% in the full-scale range, pH precision at ±1 pH, and NPK precision at 2% Full Scale. The SPEC sensor has a rapid response time of less than 1 second with an IP68 protection level. This sensor enables us to determine the exact amount of soil nutrients without going through a conventional time-consuming, and expensive laboratory testing process[[28]](#_bookmark74).

* 1. Ultrasonic sensor The ultrasonic sensor shown in Figure 39 is utilized in the product to help the robot navigate in the field through obstacle detection. The sensor consists of a transducer that serves as both an emitter and a receiver. Ultrasonic waves are emitted from the sensor and are reflected when an object in its path is detected.

*Figure 25 HC-SR04 sensor*

By measuring the time delay between the emission and reception of these echoes, and using the speed of sound in the medium, the sensor calculates the distance to that object. The working voltage of the sensor is 5V DC with a working current of

15 mA. It has a range between 2 to 400 cm with an accuracy of 3 mm.

* 1. A servomotor, or servo, is a type of rotary or linear motor that uses feedback to precisely control its position, speed, and torque. Its main modes of operation are receiving control signals and adjusting its position based on the feedback received. In this project, a servomotor serves a crucial role in the data acquisition process. The servo motor is positioned in a way so we can move the SPEC sensor towards the soil , to completely immerse the electrodes with the soil for accurate reading

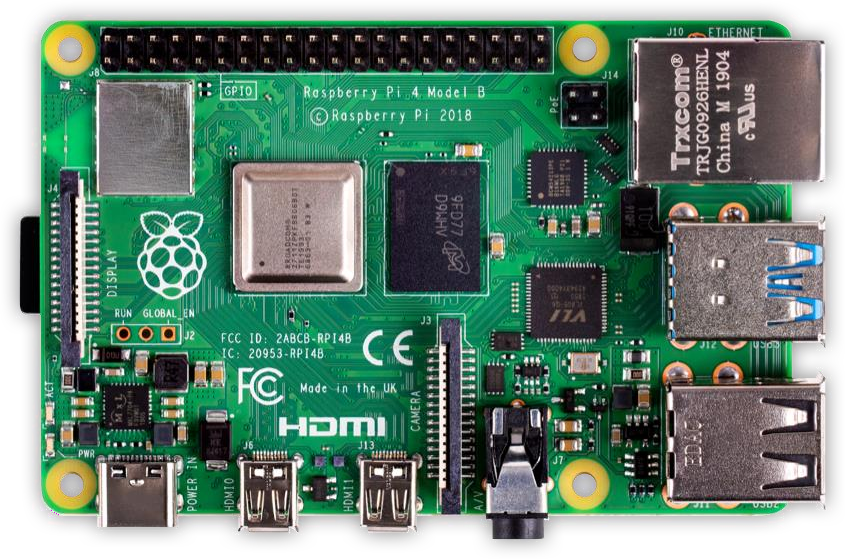
.The SG90 Micro Servo as seen in Figure 23 is a compact and lightweight device that offers a stall torque of 1.8 N.cm. It provides a rotation of approximately 180 degrees (90 degrees in each direction) and operates at a speed of 0.1 seconds for a 60-degree movement. The servo is designed to work within an operating voltage range of 4.8V to 5V, with a dead bandwidth of 10 μs..



*Figure 26 SG90 Micro Servo*

.

* 1. Raspberry Pi 4 Model B is considered a powerful computer, providing significant upgrades in performance over others models with different variants of 1GB, 2GB, 4GB, and 8GB of RAM, and the support of dual-band Wi-Fi with frequencies of 2.4 GHz and 5 GH, and support pf Bluetooth 5. It has 40 GPIO pins for interfacing many different sensors and devices—everything on a single board capable of carrying strong computing and connectivity needs.



*Figure 27 Raspberry Pi Model B*

For our project, Raspberry Pi 4 can be used in data collection and processing by connecting different sensors. It can serve as a web server for the visualization of real- time sensor data and remote supervision and control. It also has camera modules that help in taking pictures of crops for visual examination in health checks and pest infestation analysis. More importantly, Raspberry Pi can support the integration of ai model for predictive analytics and automation tasks like detect crop disease, thereby vastly improving the capability and effectiveness of the rover in agricultural management.

* 1. The Raspberry Pi Camera Module is the best choice for our rover because it has better image quality and is perfectly compatible with Raspberry Pi boards. Resolutions of up to 12 megapixels and the ability to record video provide detailed visual representations of crops, hence facilitating effective monitoring and assessment of their health. Moreover, it works well with many models of Raspberry Pi; thus, its integration in our application should be easy. To communicate with the Raspberry Pi Camera Module, one can make use of the dedicated camera interface—CSI on the Raspberry Pi—which allows for easy configuration and programming using libraries like picamera.



*Figure 28 Raspberry pi camera*

Arduino Mega:

The Arduino Mega is a powerful microcontroller board based on the ATmega2560 chip. It has an unusually high number of input/output pins: 54 digital pins, of which 15 can be used as PWM outputs, and 16 analog inputs, making it suitable for complex projects with multiple connections. The Mega has a USB connection for programming and a feature of memory capacity larger than other Arduino boards.



*Figure 29 Arduino Mega Board*

This would be particularly useful in our project when dealing with a large number of sensors and actuators at the same time using the Arduino Mega. The board would, for example, handle a number of soil moisture sensors, temperature sensors, and cameras for a complete environmental monitoring system; that is, it will enable controlling several motors or pumps for advanced functionalities. The Mega has more processing power and memory, so it's good for scenarios where higher data throughput and complex algorithms are involved, to make sure that the rover works well under different agricultural tasks.

Relay:

* 1. In our rover, a relay module acts as a critical component that allows control of high- voltage devices by using low-voltage signals from the microcontroller. Primarily, a relay is an electrically operated switch that allows the safe turning on or off of devices such as motors, pumps, or lighting systems. Moreover, the relay module will provide electrical isolation between the microcontroller and the high-voltage components, which protects the sensitive electronic components from any possible voltage spikes or surges. Most relay modules come with multiple channels, allowing the user to control many devices at once. This feature proves to be especially useful in the automation of different tasks in the rover, like switching movement controls.



*Figure 30 SRD-5VDC-SL-C Relay Module*

Linking a relay module with a microcontroller, like an Arduino or an ESP32, is not a very complex operation. The control pins of the relay are connected to the digital output pins of the microcontroller; the power and ground pins are connected to an appropriate source. The above wiring allows the microcontroller to energize the relay and hence control the high-power devices without jeopardizing its internal circuitry.

* 1. Motors :

The project will make use of DC motors, an important component for the control in the movement of our rover. Generally, DC motors are simple, yet effective, in robotics, especially for mobile platforms like ours. Each of the DC motors will drive the wheels of the rover for proper maneuverability and control.



*Figure 31 DC Motor*

* 1. Monster Moto Shield

In our rover the Monster Moto Shield used for controlling the DC motors that drive the rover. It allows the easy communication with the microcontroller and is efficient way to control the speed and direction of the motors. The Monster Moto Shield improves the responsiveness of the rover, which is quite essential in traversing through different agricultural landscapes, thanks to features such as Pulse Width Modulation (PWM) for speed control and the ability to control several motors simultaneously.



*Figure 32 Monster Moto Shield Module*

This not only makes the wiring and programming easier but also allows for fast changes and makes it highly scalable. This becomes very helpful in activities where precision movement is required, for example, turning or getting around obstacles while conducting agriculture. Our rover is equipped with reliable and efficient motion using the Monster Moto Shield with DC motors, which turns it into a powerful tool useful in modern agricultural techniques.

* 1. Rover Chassis

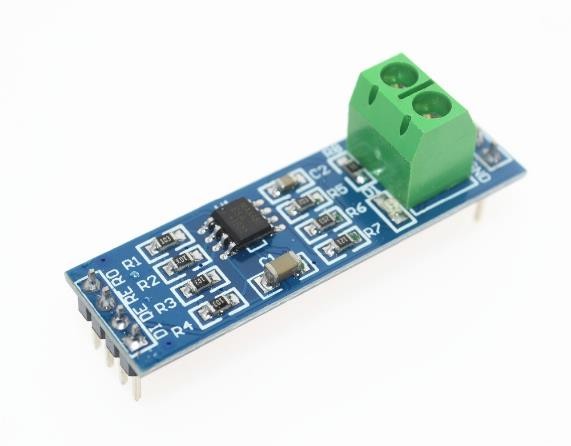
The chassis we are incorporating into our project serves as the fundamental backbone for our rover prototype



*Figure 33 Rover's Wheeled Chassis*

* 1. RS-485

RS-485 used with the NPK sensor will enhance data transmission capabilities to provide long-distance communication of up to 4,000 feet with no significant degradation of signal strength. This protocol is of considerable importance in agricultural environments, where electrical interference usually exists, due to its high noise immunity and support for multipoint communication, which allows many sensors to be wired on the same bus.



*Figure 34 RS-485 Module*

The advantages of using RS-485 with the NPK sensor include increased data integrity, reduced wiring complexity, and improved system flexibility. We can scale the system easily by adding or removing sensors as needed because we reduce the complexity of connections. In addition, the low cost of RS-485 components, together with their ruggedness for outdoor use, makes this communication method quite economical, enhancing the rover's ability to manage soil nutrient levels effectively for better agricultural practices.

### Feasibility Study

A feasibility study is one of the major steps involved in assessing the possibility and viability of any project. It helps us identify the strengths and weaknesses of our proposed system, opportunities and threats present in the natural environment, resources needed to carry through, and finally the prospects of success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. In the case of agricultural robot project, the feasibility study addressed the following:

*Table 12 Feasibility Study*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Quantity | Price per unit | Total Price | Weight |
| Arduino Mega | 1 | 45$ | 45$ | 37g |
| Raspberry 4 Model B | 1 | 120$ | 120$ | 46g |
| Raspberry Pi Camera Module | 1 | 35$ | 35$ | 20g |
| Relay | 1 | 3$ | 3$ | 20g |
| MOTOR DRIVER | 1 | 5$ | 3$ | 50g |
| DC Motor | 4 | 15$ | 15$ | 400g |
| Rover Chassis | 1 | 50$ | 50$ | 1000g |
| Rs-485 | 1 | 8.99$ | 8.99$ | 10g |
| Spec 7 in 1 sensor | 1 | 48.11$ | 48.11$ | 50g |
| Total |  |  | 330.1$ | 1,633g |

The technical Feasibility involves the assessment of the availability and suitability of the required technologies like sensors, navigation systems, and software. This ensures that all conditions regarding sufficiency and availability of the required technologies for the functionality of the robot are attained within the guidelines of the projected scope. Estimating costs is one of the first steps to determine whether the project is financially feasible. It is involving the consideration of expenses concerning hardware acquisition,

development, and maintenance, along with unforeseen costs of upgrading to improve programming efficiency.

A cost-benefit analysis will be necessary for confirming that the investments agree with the expected outcomes. Operational Feasibility is another key aspect involves assessing the efficiency of the robot in the field, which means how well it can function on various grounds and climatic conditions and how efficiently it can be operated and maintained to meet its requirements without much operational difficulty.

In the world of IoT, the IoT devices are interconnected among themselves and with applications using communication protocols, which allow for the flow of data and information. There exists a gap between the IoT devices and the IoT applications themselves. An IoT platform bridges this gap by providing an integrated service that enables devices and applications to communicate, thus bridging the physical world of sensors and the digital world of software applications. Such platforms offer key services that include device management, security, real-time data processing, and analytics, among others, which make them an essential component of IoT systems. In the case of the agricultural robot, an IoT platform connects sensors and devices, manages various software communication protocols and hardware, ensures security and authentication, and provides tools to collect, visualize, and analyze data from the robot's sensors. The choice of platform will depend on the project's specific requirements: budget, scalability, data storage needs, and integration of machine learning or AI. AWS and GCP are considered the primary options for the agriculture robot project because of their comprehensive services and global infrastructure. However, competitive solutions could also be provided by alternatives like Microsoft Azure, IBM Cloud, Thing Worx Cloud, Cloud, especially for enterprise applications, AI-focused projects, or particular geographic needs.

AWS IoT Platform

AWS IoT incorporates a broad set of cloud services that allow the connected things or IoT devices to easily and securely interact with the AWS cloud services. It features robust security and communication capabilities, thus making it perfect for large-scale IoT applications. AWS IoT Core, as its central component, allows the use of several protocols-MQTT, HTTPS, and Lora WAN-to securely and in real time enable devices and the cloud to interact with one another. AWS IoT also provides a device shadow, where the device state is kept in the cloud even when it may be disconnected. Besides all that, AWS IoT will provide great scalability and flexibility since it is integrated with many other AWS services, such as serverless computing with AWS Lambda and data storage with Amazon S3. It also offers services like device management, message routing, and data processing, thus making it fit for complex IoT solutions, such as the agriculture robot project.

## Phase 3 :Website Development

During this phase , we will focus on developing a website that will provide the primary interface from the sensor (soil and camera ) sensors that helps farmers and land owners have some insights about their crop health .The microcontroller manages this data coming from different sensors connected to the rover for transmission and visualization for analysis to a website. It would, the microcontroller process real-time data through sensors along with different environmental conditions. Building upon pervious phases The website interprets and provide visualization to the data captured by the rover, and can provide recommendations according to the data captured .

The various design alternatives and options that were considered for the website, including tools chosen to construct it, will be discussed in this stage. How the website actually connects to the microcontroller and enables the rover to communicate to the web interface in real time and vice-versa.

#### Optimum Selection

In this section, after viewing some alternative designs we evaluate the best tools that meets our aim and objectives of this project .The website will be supporting the user interface, assess solutions for robust data processing, and ensure reliable data exchange between the rover and the web application. We also needed to develop database systems efficient for data storage and retrieval, and cloud platforms regarding scalability and integration.

### Frontend:

We selected React as a frontend framework because of its easy interfacing and its component-based design which supports building of reusable and maintainable code. React not only enhances the user experience by facilitating the perfect user interface for monitoring and analysis, but it also offers the capability of embedding various visual elements, in providing an easier understanding of such complex data patterns related to soil moisture levels and nutrient contents. This is crucial because farmers and agricultural researchers do have to take prudent and timely actions by all means, based on the insights harnessed from this visualization. Towards the back end, react lends itself quite adequately to allowing other integrations of interests, such as real-time communications protocols, which could be implemented in providing instantaneous feedbacks regarding the status about soils. Many of Reacts’ capabilities go a long way in

powering and user friendliness for our web frontend; hence, crucial to Precise Agriculture as a company.

### Backend:

In our project, where we handle the rapid feedback that is important for tasks such as crop adjustments based on soil sensor reading, diagnosing plant diseases or ripeness of crops through AI/ML analysis, Flask holds it ahead. Flask is a lightweight web framework, and this will be our backbone for our project on the backend, thus enabling us to develop flexibly in a simple way. This handles HTTP requests efficiently, processes data from the frontend, and manages interactions with our MongoDB database for real-time access to metrics in the soil and also enables users to access their historical records. This application is useful for informed decision-making at farm level because it will easily allow for a comparison between current sensor and past sensor readings. We are also going to include an appropriate AI model integrated using Flask, which will review images taken by the rover, enhancing our understanding of the health status of the crops and which one has attained ripeness. Having in mind all the strong points of Flask, besides its simple architecture, it will be ideal for developing a responsive and efficient backend that allows our application to meet the demands in precision agriculture.

### Communication Protocol:

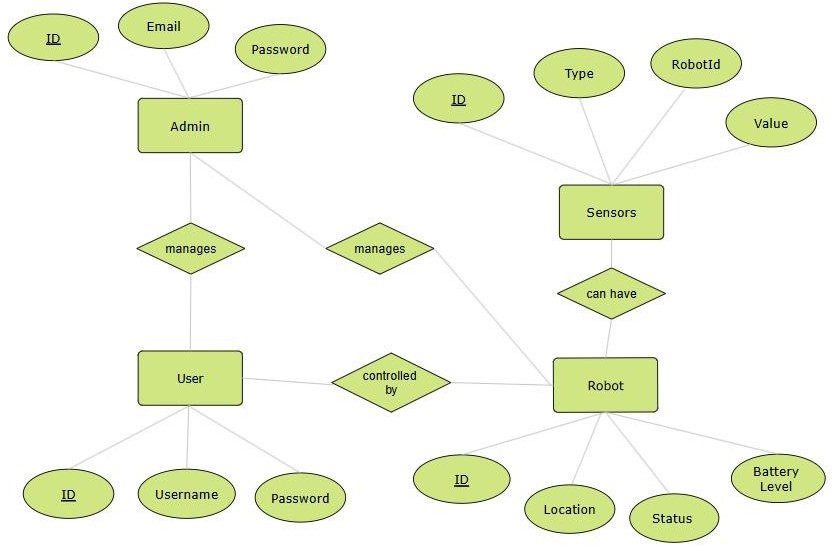
While MQTT and WebSocket have different purposes, their combination gives great results. By wrapping MQTT over WebSocket, developers can exploit the robust messaging system of MQTT within web environments and allow IoT data to be seamlessly integrated into web applications. This enables direct interaction with IoT devices through web browsers in real time and increases user experience, thereby extending IoT capabilities to the broader web.

Using MQTT with WebSocket in our agriculture project it will combine the advantages of both in order to improve the real time communication of your system. The MQTT uses the publish/subscribe data transfer that is the right fit for low bandwidth and it will be suitable for transmitting data from the robot’s sensors and cameras to the monitoring dashboard. When integrated with WebSocket which provides bidirectional communication users can receive update notifications at the same time as they are sending commands to the robot. This synergy guarantees data integrity, robustness, and security, which makes such systems useful in agricultural environments where

connectivity could be a challenge or intermittent, yet timely information is important approaching decision-making.

### Database:

MongoDB, NoSQL database, enables us to handle the unstructured data from the sensors and cameras connected to rover. It can store and process sensor data in real time, thus enabling efficient analysis and visualization of the information gathered. In this respect, the document-based model of MongoDB plays a very important role due to frequent changes in schemas, especially when one deals with various data provided by the robot, whereas a conventional relational database does not work out in these conditions. Moreover, with this ability to transform unorganized raw data into organized forms, the aforementioned adaptability will help to derive actionable insights from it, thus providing support for the creation of complex applications. Besides, the handling of structured and semi-structured data by NoSQL adds to artificial intelligence's power. MongoDB becomes an ideal solution to solve our agricultural monitoring system and the demanding feature set required thereof for better decision-making and operational efficiency.



*Figure 35 ER Diagram*

The ER diagram shown in Figure 35, depicts the integration of users, rovers, sensor data, images, user issues, admin, and alerts entities. Entity User is the foundation entity of the system, which is defined with respect to a uniquely defined UserID that allows each individual to maintain his or her respective account. The attributes involved are Username, Password, and Email. User Issues allow the user to raise issues. Each issue description is kept along with a timestamp. Admin: It is with respect to one AdminID, which enables the admin to resolve issues reported by any user and control the particular user's account. Rover: A robot collecting sensor data, and images with their respective IDs as RoverID. Sensor Data: The readings, which are dispatched by each rover, subsume many numeric units: moisture, nutrient levels;. Also, the rovers capture images that are stored with field names like Image URL, and Analysis Result in the entity Images.This design emphasizes, in general, the interaction within the system: from real- time or historical data views from sensors and cameras to control by users over the rover. Issues will also be managed by an admin, who collaborates with the robots to provide important environmental information for better monitoring and decision-making.

### Cloud Platforms

We use Amazon Web Services in the agriculture rover project since, owing to the high demand on scalability, security, and integration of the data used in our architecture of Flask-React-MongoDB, AWS supports all three ends. AWS offers its services for Flask on the back-end, for instance, by Elastic Beanstalk; deployment is pretty smooth and effortless, while it's automatic scaling is handled regarding the demand. AWS also provides content delivery via Amazon S3 and CloudFront to ensure fast loading for our React frontend. We also lean on AWS Document-DB as a managed service for handling our MongoDB needs, thus freeing our team from maintenance. Consequently, with AWS, one will be able to apply full-featured CPU choices together with scalable storage alternatives for hosting the application in response to variable workloads to maintain high performance and dependability. AWS further streamlines the handling of AI models in data analysis and decision-making, hence allowing data processing to occur in real time-as obtained from the rover. Further, this allows us to apply machine learning algorithms to analyze sensor data on-the-fly for instant insights and recommendations into agricultural practices that will enhance the general functionality and responsiveness of our system.

### Website Overview

The development of our project’s website required the evaluation of several technology options in order to design a base that is both strong and effective for a web-based system. Out of the several options, we have finally selected Flask for the back end, React for the front end, and MongoDB for data storage and serving all of which are to be hosted on AWS. All these platforms not only meet the needs for real-time data capture and processing needed in agricultural management but also provide scalability and security as we increase the number of users and the amount of data being stored.

The website is a crucial part of our agricultural robot project. This web interface allows the user to visualize and analyze the data collected by the rover with regard to the soil on which it moves. It not only allows access to real-time soil conditions but also brings in AI for crop ripeness assessments and leaf disease detection. It helps the farmers by using these functionalities to make decisions for better management and more sustainability of crops.

The main part of our website is the real time data from the rover. To achieve this, the rover has been fitted with several sensors that measure key soil properties like moisture content, pH level, temperature and nutrient uptake. This data is then sent through controllers to our website as RESTful API which allows for easy connection between the rover and the server. This is where we have chosen Flask as our backend framework that helps us in managing these API requests in an efficient manner and thus ensuring that the data received is always real time and acted upon without delay.

When the data gets to the server it is first validated and then processed in order to be saved into our MongoDB database. This NoSQL database is suitable for diverse data sets collected by the sensors and it also has time series data which plots changes over time and is important in analyzing variations that affect the agriculture sector. Also, the rover’s controllers are responsible for managing the transmission of other data such as the camera and the sensors, thus providing real time images and metrics on the website for further examination.

Our website is designed to provide the user with a comprehensive overview of the condition of the soil through the use of analytics. We give insights that help users make informed decisions using algorithms that analyze current and historical data. For example, if the soil moisture has been low over a given period, recommendations are given on fertilizers practices.

We improved the user experience with interactive graphical and visualization of current and historical data. Using libraries inside our React frontend, we present data in a friendly, readable manner. With these charts, users can easily zoom in into a specific window of time or metric and dive more deeply into the visualizations-very important for necessary adjustments of farming practices in a timely fashion.

Upon logging to the website, users are greeted with an dashboard designed using React, which allows users to view real-time data visualizations, including interactive graphs that display variations in soil nutrient content (N-P-K) and camera results from rover. Every interaction made by a user had been planned for the exact capturing of necessary information in accessible formats.

Once the users inside the dashboard, they can see data captured by the rover in real time. The rover sends all the information collected from the soil sensor and high-resolution images through to the website in order to stream them as live feeds, while assessments of the ripeness of crops or the health of leaves come in through AI model that is integration to the website through backend that is powered by Flask, which allows for rapid development and easy integration with various data sources. It efficiently handles incoming data from the rover, processes requests from users, and interacts with MongoDB to store and retrieve information, ensuring smooth communication between the frontend and backend. This will give immediate access to critical information so that they can make a decision in time regarding their crops and the way of farming.

The website allows users to deep dive into the historical data stored in the robust MongoDB database. This allows them to go back through previous readings and compare those with current data for extended analysis of trends over time. Based on how the soil conditions have changed, one will be able to develop an optimized farming strategy from the historical performance.

Also, whenever the rover captured the images or video streaming, the AI tries to predict the ripeness and potential diseases in crops from them. These analyses, along with the raw images, are then forwarded to the user to help in making informed decisions related to harvesting and management.

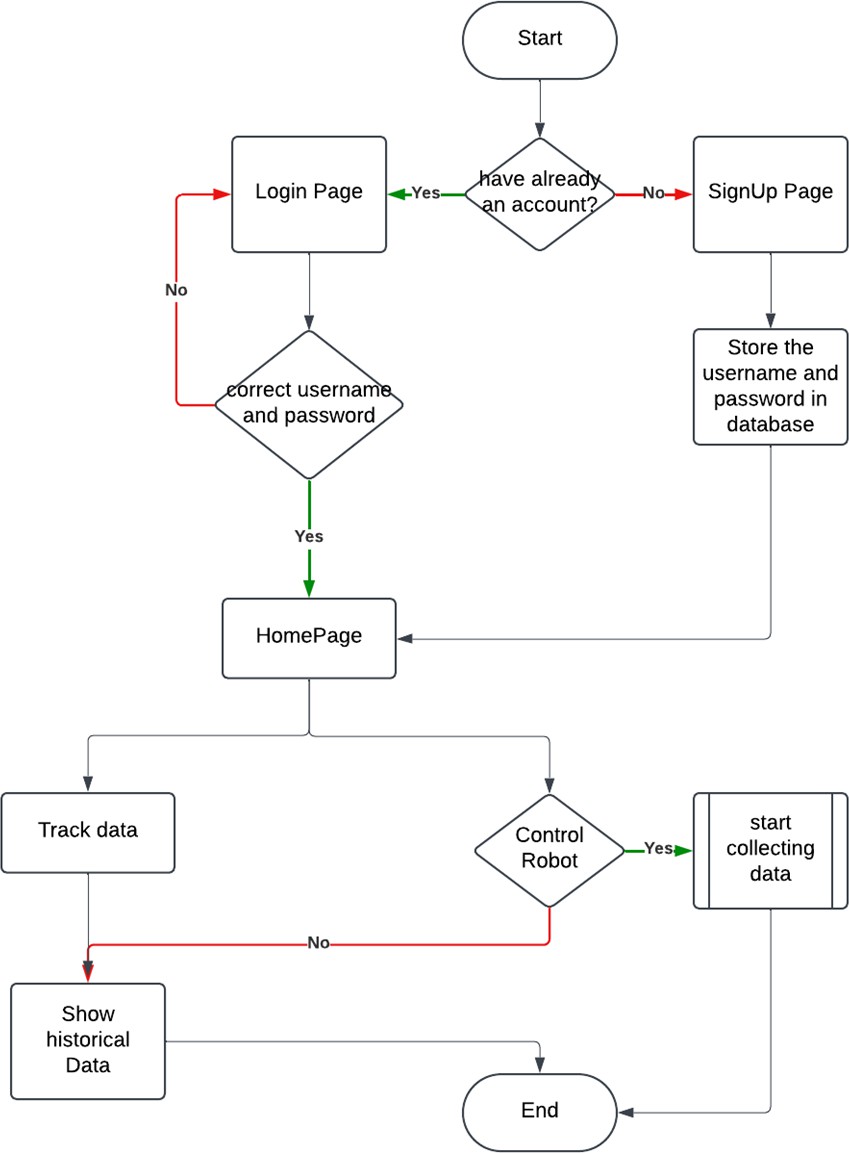
MQTT and WebSocket provide the basis for real-time alerts on the website. MQTT enables light messaging, which is fast in transmitting data, thus allowing the timely delivery of soil sensor data. On the other hand, WebSocket enables bidirectional communication; hence, users get immediate updates on the changes in soil conditions and crop health, which is critical in proactive decision-making.

Us as administrators we play an important role in keeping the backend current. We ensure proper structuring and organization of data input into MongoDB; data will be recorded correctly and retrievable with ease. In addition, we are supposed to monitor general website health, fix problems as they arise, and implement updates for user experience. They also offer support to users by helping them with questions or technical issues so that the users can reap maximum benefit from the platform.

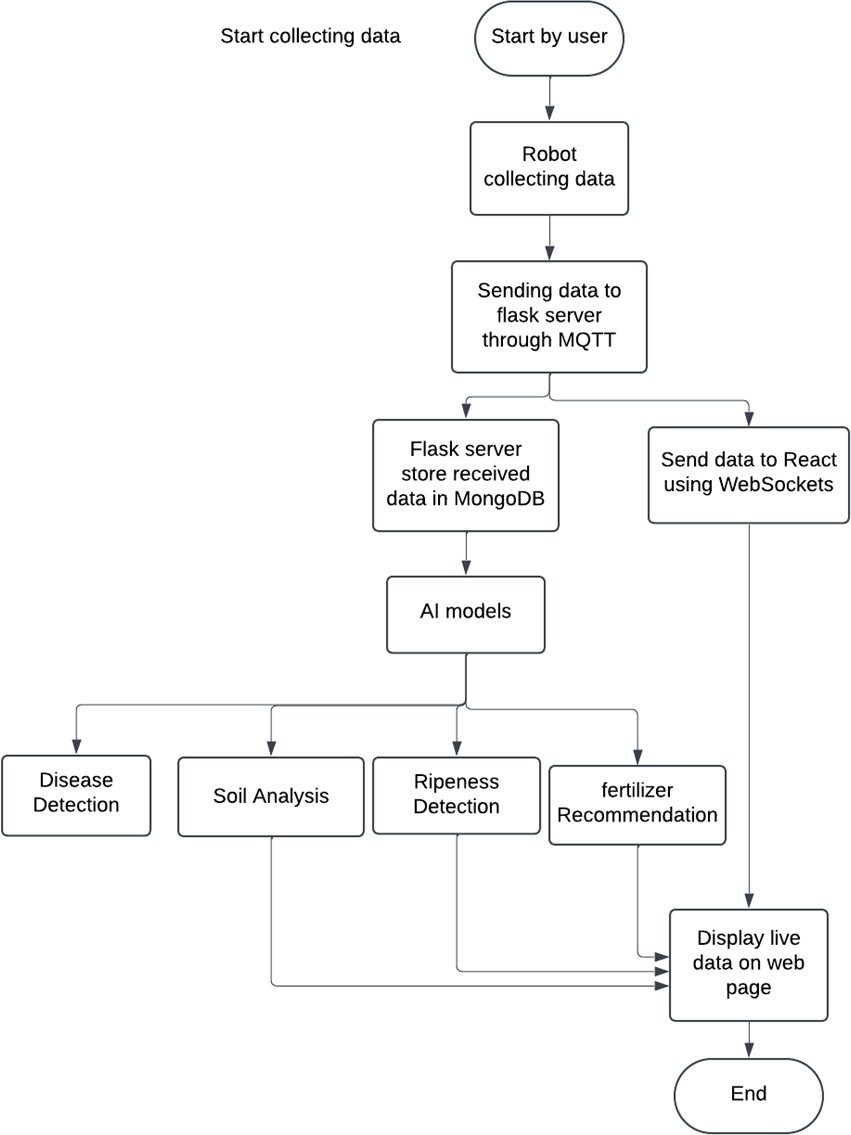
By deploying our web application on AWS, we provide our project with a host infrastructure that is both dependable and scalable. AWS brings the tools to manage these hosting needs, which consist of Elastic Beanstalk to deploy the application and S3 for storing the image taken by the rover camera. This cloud-based solution ensures high availability and scaling in the future, especially when the user base is increased and data volume goes up. Thus, we can use AWS to build in security and allow the user's data to remain safe. Regular backup and monitoring services keep our platform intact and available to stay focused on continuous improvements and expanding features.

Our website acts as the important interface through which the user is connected to the sophisticated functionalities of our Agri-robot project. With real-time data collection, enriched analytics, and a user-friendly design, agricultural professionals can optimize their practices to attain higher crop productivity. It is because of this that the various thoughtful integrations of modern technologies lie at the heart of our commitment to improvement through innovation in agricultural outcomes. Refining the platform and continuing its expansion, we remain attached to supporting users in their search for sustainable agricultural practices, in such a way as to enable a healthier future for crops and the environment.

The following figure represents the flow chart of our system architecture; this shows the flow from data gathering on the rover through to data processing and the handling of the website's backend and user interaction.



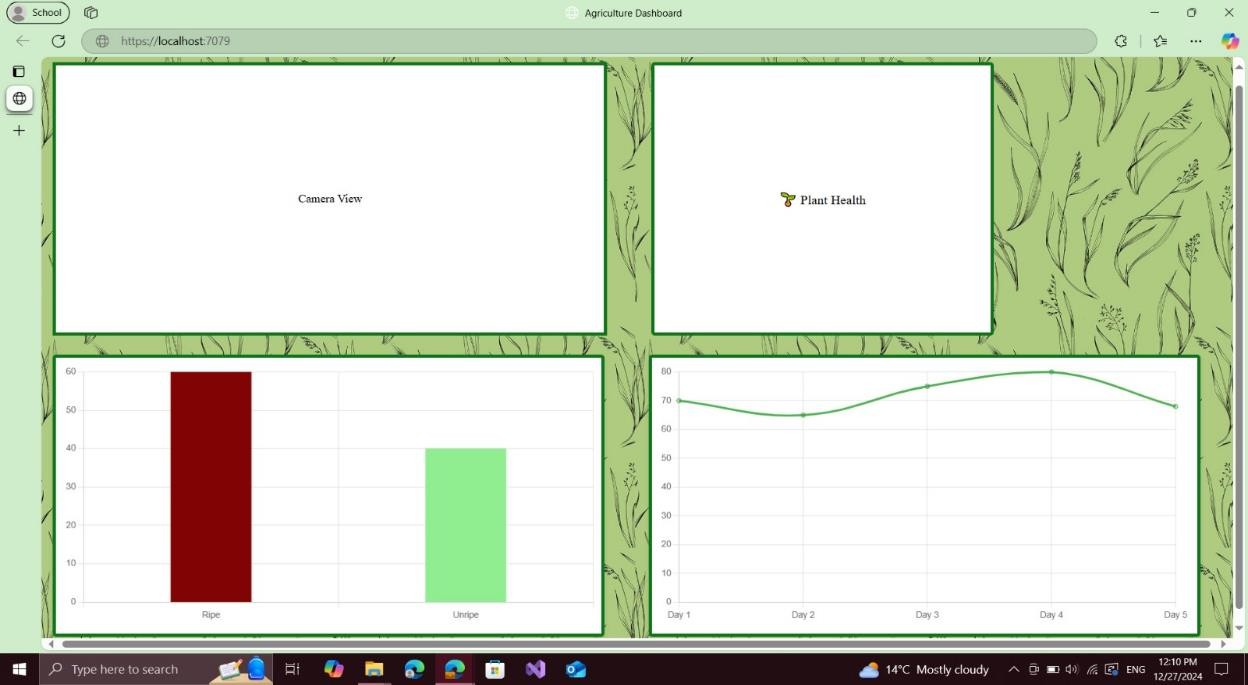
*Figure 36 Users Accessibility Flowchart*



*Figure 37 Rover and Website Relationship Flow Chart*

# Design

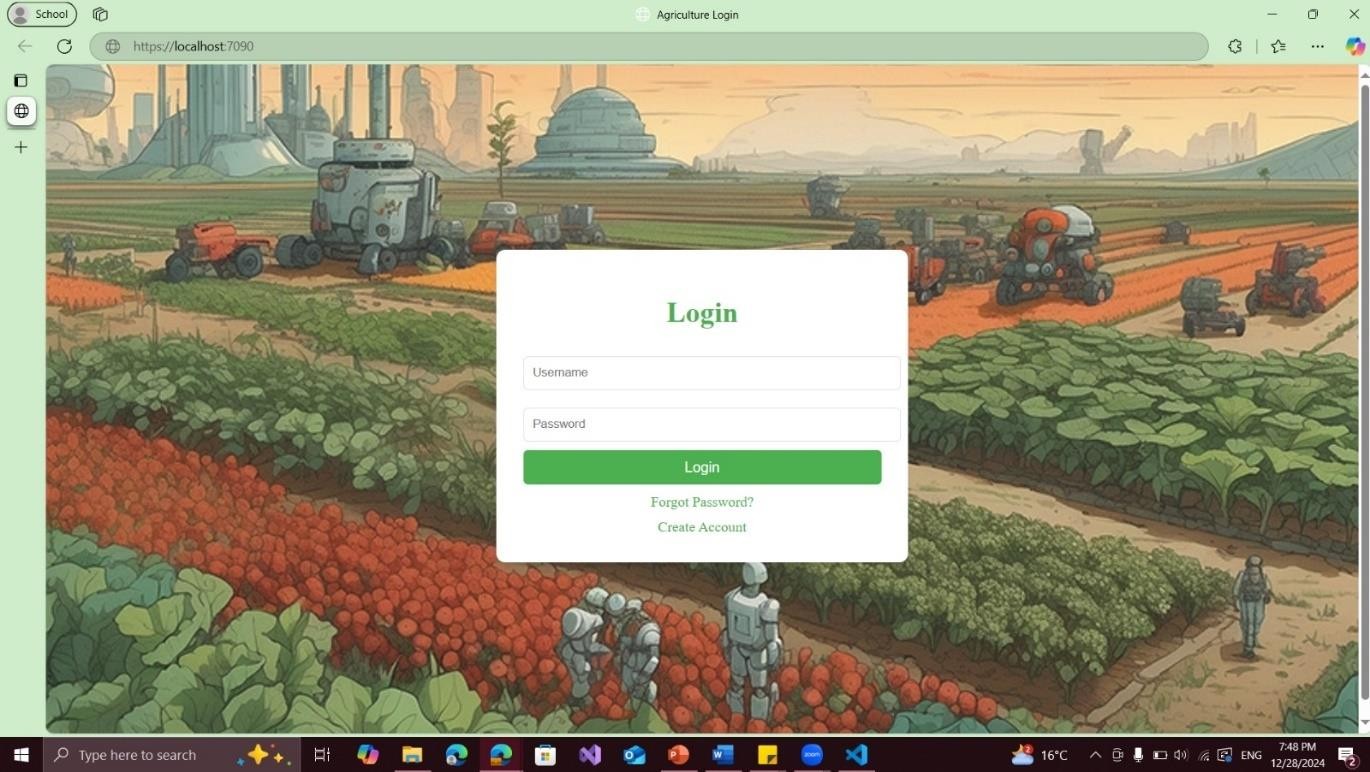
Once data is collected from soil sensor and camera, it is crucial that this information is presented in a clear and engaging manner where the website frontend should be user friendly to fit our users of this project which are mainly farmers and landowners and most of them aren’t engaged with technology a user-friendly frontend help them in getting insights without any difficulties or challenges . The website interface is designed to display the data captured in visually compelling formats like the histogram and graph shown below :



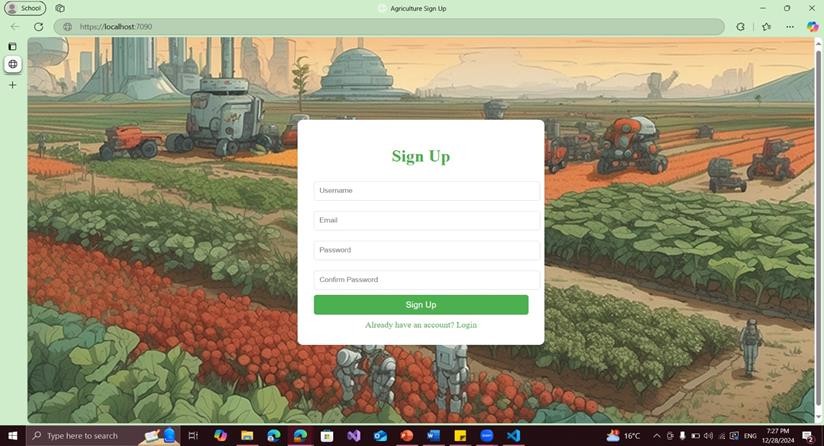
*Figure 38 Home page*

As shown above in figure 38 where it presents the home page of the user presenting the data captured in friendly visuals.

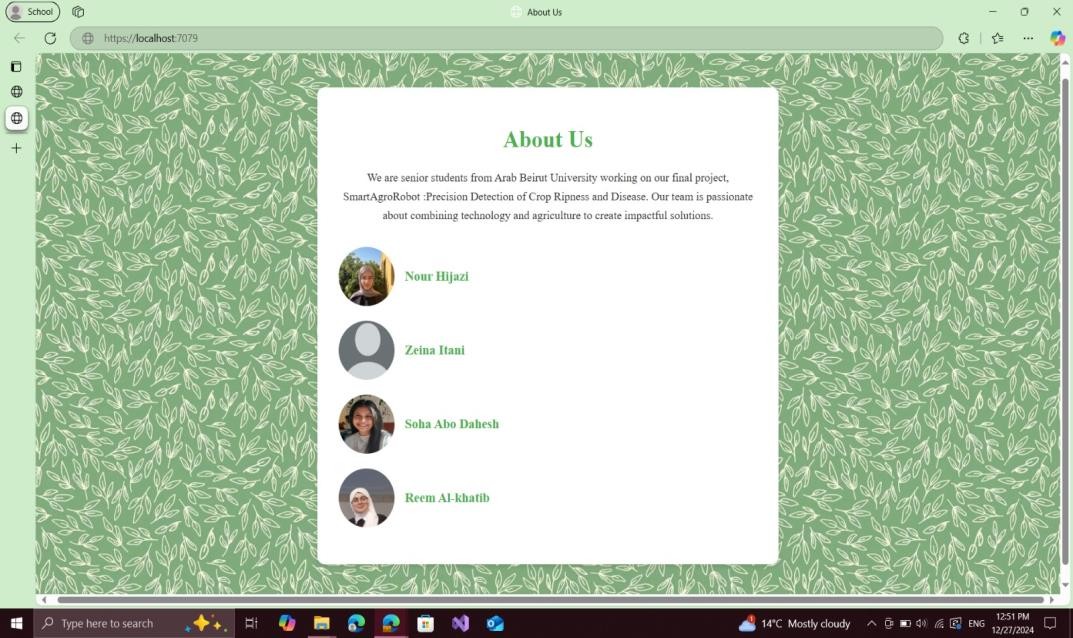
The below images show sign up , login and about us pages respectively :



*Figure 39 Login Page*

**

*Figure 40 Sign Up Page*



*Figure 41 About Us Page*

## Bibliography:

1. Zhang, B., & Qiao, Y. (2024). AI, Sensors, and Robotics for Smart Agriculture. Agronomy, 14(6), 1180. <https://doi.org/10.3390/agronomy14061180>
2. Dr. Hadi Jaafar. (2024). The 2024 Israeli War on Lebanon: A Devastating Blow to Agriculture and the Environment [online]. Retrieved December 21, 2024, from American University of Beirut website: [https://www.aub.edu.lb/fafs/news/Pages/2024\_The-2024-Israeli-War-on-Lebanon-A-](https://www.aub.edu.lb/fafs/news/Pages/2024_The-2024-Israeli-War-on-Lebanon-A-Devastating-Blow-to-Agriculture-and-the-Environment.aspxin) [Devastating-Blow-to-Agriculture-and-the-Environment.aspxin](https://www.aub.edu.lb/fafs/news/Pages/2024_The-2024-Israeli-War-on-Lebanon-A-Devastating-Blow-to-Agriculture-and-the-Environment.aspxin)
3. Mac, T. T., Nguyen, T. D., Dang, H. K., Nguyen, D. T., & Nguyen, X. T. (2024). Intelligent agricultural robotic detection system for greenhouse tomato leaf diseases using soft computing techniques and deep learning. *Scientific Reports*, *14*(1), 23887.
4. Apostolopoulos, Ioannis D., Mpesi Tzani, and Sokratis I. Aznaouridis. 2023. "A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features" AI 4, no. 4: 812-830. <https://doi.org/10.3390/ai4040041>
5. Jiajun Li, Zifeng Zhu, Hongxin Liu, Yurong Su, Limiao Deng,Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R- CNN,EcologicalInformatics,Volume77,2023,102210,ISSN15749541,https://doi.org/10.1016/j.eco inf.2023.102210 .
6. Zhao, J., Du, C., Li, Y. et al. YOLO-Granada: a lightweight attention Yolo for pomegranates fruit detection. Sci Rep 14, 16848 (2024). <https://doi.org/10.1038/s41598-024-67526-4>
7. Wang, C., Wang, C., Wang, L., Wang, J., Liao, J., Li, Y., & Lan, Y. (2023). A Lightweight Cherry Tomato Maturity Real-Time Detection Algorithm Based on Improved YOLOV5n. Agronomy, 13(8), 2106. <https://doi.org/10.3390/agronomy13082106>
8. Pickett, R. A., Nowlin, J. W., Hashem, A. A., Reba, M. L., Massey, J. H., & Alsbrook, S. (2023). Small Unmanned Aircraft Systems and Agro-Terrestrial Surveys Comparison for Generating Digital Elevation Surfaces for Irrigation and Precision Grading. Drones, 7(11), 649. <https://doi.org/10.3390/drones7110649>
9. Drones for Spraying Pesticides—Opportunities and challenges. (n.d.). Ohioline. <https://ohioline.osu.edu/factsheet/fabe-540>
10. Das B, Sayor TZUH, Nijhum RJ, Tishun MT, Sakib TH, Karim ME, Uddin AJ, Islam A, Mohsin ASM. Designing and development of agricultural rovers for vegetable harvesting and soil analysis. PLoS One. 2024 Jun 21;19(6):e0304657. doi: 10.1371/journal.pone.0304657. PMID: 38905232; PMCID: PMC11192377.
11. Jothilakshmi, R., & Sharanesh, R. (2020). Automated Plant Disease Detection using Deep Learning Architectures with Autonomous rover. Int J Recent Technol Eng, 9(2), 248-254.
12. Karim, M.J., Goni, M.O.F., Nahiduzzaman, M. et al. Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM. Sci Rep 14, 16022 (2024). <https://doi.org/10.1038/s41598-024-66989-9>
13. Ahmed, R., & Abd-Elkawy, E. H. (2024). Improved Tomato Disease Detection with YOLOv5 and YOLOv8. Engineering, Technology & Applied Science Research, 14(3), 13922-13928.
14. Mac, T.T., Nguyen, TD., Dang, HK. et al. Intelligent agricultural robotic detection system for greenhouse tomato leaf diseases using soft computing techniques and deep learning. Sci Rep 14, 23887 (2024). <https://doi.org/10.1038/s41598-024-75285-5>
15. H. R and V. K. N, "Insights on Assessing Image Processing Approaches Towards Health Status of Plant Leaf," 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2022, pp. 1-7, doi: 10.1109/ICSTCEE56972.2022.10099507.
16. Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., & Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS journal of photogrammetry and remote sensing, 145, 3-22.
17. Keylabs. (2024, February 27). YOLOv8 vs Faster R-CNN: A Comparative Analysis. Keylabs: Latest News and Updates. <https://keylabs.ai/blog/yolov8-vs-faster-r-cnn-a-comparative-analysis>
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.
19. V. K. Kaliappan, M. S. V, K. Shanmugasundaram, L. Ravikumar and G. B. Hiremath, "Performance Analysis of YOLOv8, RCNN, and SSD Object Detection Models for Precision Poultry Farming Management," 2023 IEEE 3rd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), Bhubaneswar, India, 2023,

pp. 1-6, doi: 10.1109/AESPC59761.2023.10389906

1. Keylabs. (2024, February 27). YOLOv8 vs SSD: Choosing the Right Object Detection Model. Keylabs: Latest News and Updates. [https://keylabs.ai/blog/yolov8-vs-ssd-choosing-the-right-](https://keylabs.ai/blog/yolov8-vs-ssd-choosing-the-right-object-detection-model/) [object-detection-model/](https://keylabs.ai/blog/yolov8-vs-ssd-choosing-the-right-object-detection-model/)
2. Potrimba, P. (2024, April 15). What is Mask R-CNN? The Ultimate Guide. Roboflow Blog. <https://blog.roboflow.com/mask-rcnn/>
3. Sapkota, R., Ahmed, D., & Karkee, M. (2024). Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. Artificial Intelligence in Agriculture, 13, 84-99.
4. Kirouane, A. (2023, March 10). A Step-by-Step Guide to Implementing RetinaNet for Object Detection using Keras and Detectron2. [https://www.linkedin.com/pulse/step-by-step-guide-](https://www.linkedin.com/pulse/step-by-step-guide-implementing-retinanet-object-using-ayoub-kirouane) [implementing-retinanet-object-using-ayoub-kirouane](https://www.linkedin.com/pulse/step-by-step-guide-implementing-retinanet-object-using-ayoub-kirouane)
5. Butt M, Glas N, Monsuur J, Stoop R, de Keijzer A. Application of YOLOv8 and Detectron2 for Bullet Hole Detection and Score Calculation from Shooting Cards. AI. 2024; 5(1):72-90. <https://doi.org/10.3390/ai5010005>
6. Ultralytics. (2024, November 7). YOLOV8. Ultralytics YOLO Docs. <https://docs.ultralytics.com/models/yolov8/>
7. Luo, J., Pan, Y., Su, D., Zhong, J., Wu, L., Zhao, W., ... & Wang, Y. (2024). Innovative cloud quantification: deep learning classification and finite-sector clustering for ground-based all-sky imaging. Atmospheric Measurement Techniques, 17(12), 3765-3781.
8. S. Dattatreya, A. N. Khan, K. Jena and G. Chatterjee, "Conventional to Modern Methods of Soil NPK Sensing: A Review," in IEEE Sensors Journal, vol. 24, no. 3, pp. 2367-2380, 1 Feb.1, 2024, doi: 10.1109/JSEN.2023.3334243.
9. Ndidiamaka, O. (2024b, November 11). Comparing Frontend Frameworks: Angular vs. React vs. Vue.js - Which One Fits Your Project Best? DEV Community. [https://dev.to/okoye\_ndidiamaka\_5e3b7d30/comparing-frontend-frameworks-angular-vs-react-](https://dev.to/okoye_ndidiamaka_5e3b7d30/comparing-frontend-frameworks-angular-vs-react-vs-vuejs-which-one-fits-your-project-best-5epm#%3A~%3Atext%3DVue%20generally%20tends%20to%20be%2Cprojects%20where%20stability%20is%20desired) [vs-vuejs-which-one-fits-your-project-best-](https://dev.to/okoye_ndidiamaka_5e3b7d30/comparing-frontend-frameworks-angular-vs-react-vs-vuejs-which-one-fits-your-project-best-5epm#%3A~%3Atext%3DVue%20generally%20tends%20to%20be%2Cprojects%20where%20stability%20is%20desired) [5epm#:~:text=Vue%20generally%20tends%20to%20be,projects%20where%20stability%20is%2](https://dev.to/okoye_ndidiamaka_5e3b7d30/comparing-frontend-frameworks-angular-vs-react-vs-vuejs-which-one-fits-your-project-best-5epm#%3A~%3Atext%3DVue%20generally%20tends%20to%20be%2Cprojects%20where%20stability%20is%20desired) [0desired.](https://dev.to/okoye_ndidiamaka_5e3b7d30/comparing-frontend-frameworks-angular-vs-react-vs-vuejs-which-one-fits-your-project-best-5epm#%3A~%3Atext%3DVue%20generally%20tends%20to%20be%2Cprojects%20where%20stability%20is%20desired)
10. Simplilearn. (2024, August 13). Django vs. Flask: Understanding the major differences. Simplilearn.com. <https://www.simplilearn.com/flask-vs-django-article>
11. Craggs, I. (2022, July 20). Understanding the Differences between MQTT and WebSockets for IoT. HIVEMQ. Retrieved July 20, 2022, from [https://www.hivemq.com/blog/understanding-the-](https://www.hivemq.com/blog/understanding-the-differences-between-mqtt-and-websockets-for-iot/) [differences-between-mqtt-and-websockets-for-iot/](https://www.hivemq.com/blog/understanding-the-differences-between-mqtt-and-websockets-for-iot/)
12. Smallcombe, M. (2024, February 15). SQL vs NoSQL: 5 Critical Differences. Integrate.io. <https://www.integrate.io/blog/the-sql-vs-nosql-difference/>
13. Staff, C. (2024b, October 1). What’s the Difference Between AWS vs. Azure vs. Google Cloud? Coursera. [https://www.coursera.org/articles/aws-vs-azure-vs-google-](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB) [cloud?utm\_medium=sem&utm\_source=gg&utm\_campaign=B2C\_EMEA coursera\_FTCOF\_ca](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB) [reer-academy\_pmax-multiple-audiences-country-multi-](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB) [set2&campaignid=20882109092&adgroupid=&device=c&keyword=&matchtype=&network=x&](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB) [devicemodel=&adposition=&creativeid=&hide\_mobile\_promo&gad\_source=1&gclid=Cj0KCQi](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB) [A1Km7BhC9ARIsAFZfEItkYSbYH-](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB)

[sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8\_iKPMiybKFYaAp34EALw\_wcB](https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA__coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid&device=c&keyword&matchtype&network=x&devicemodel&adposition&creativeid&hide_mobile_promo&gad_source=1&gclid=Cj0KCQiA1Km7BhC9ARIsAFZfEItkYSbYH-sdMyapcPH3hXzJKuyLksCRK6cfd9vSfdd8_iKPMiybKFYaAp34EALw_wcB)