Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Carthage
Ecole Polytechnique de Tunisie

وزارة التعليـم العالي و البحث العلمي
جامعـة قرطاج
المدرسـة التونسية للتقنيات

Project Report

Elaborated by:

Yasmine Ben Mechlia
Sahar Jarboui
Aya Ben Henda
Jawhar Frih
Mohamed Bouaziz
Third Year Student at Ecole Ploytechnique de Tunisie

our partner:

La Ferme Agricole Tuniso-Suisse

Academic Year
2024 - 2025

Contents

1 Overview and Objectives 5
- 1.1 Overview 5
- 1.2 Objectives 5

2 Solution Description and Purpose 5
- 2.1 Solution Description 5
- 2.2 Purpose 6

3 Objectives and Expected Outcomes 6
- 3.1 Algorithm Development 6
- 3.2 Enhanced Disaster Response 7
- 3.3 Community Resilience 7

4 AI Part 8
- 4.1 Dataset 8
  - 4.1.1 Data Sources 8
  - 4.1.2 Data Formats 8
  - 4.1.3 Dataset Overview 8
  - 4.1.4 Class Distribution 9
- 4.2 Data Modeling 10
  - 4.2.1 Description of Data Models 10
  - 4.2.2 Model 10
  - 4.2.3 Model Training 11
  - 4.2.4 Model Validation and Testing 11
- 4.3 Performance Metrics 12
  - 4.3.1 Model Accuracy 12
  - 4.3.2 Reported Scores 13
- 4.4 Inference 14
  - 4.4.1 Input and Output of New Data 14
  - 4.4.2 Input Data 14
  - 4.4.3 Output Interpretation 14
- 4.5 Example of Predictions from the Model 15

5 Hardware Architecture for Autonomous Drones 16
- 5.1 Introduction 16
  - 5.1.1 Overview of the System 16
  - 5.1.2 Role of ROS2 16
  - 5.1.3 Gazebo for Simulation 16
- 5.2 Hardware Components 16
- 5.3 ROS2 Architecture 18

5.4 ROS2 Nodes: Inputs, Outputs, and Roles 19
5.4.1 GPS Node 19
5.4.2 IMU Node 19
5.4.3 Lidar Node 19
5.4.4 YOLO11 Node (AI-based Crop Disease Detection) 20
5.4.5 Path Planning Node 20
5.4.6 Navigation Node 20
5.5 Gazebo Simulation Overview 21
5.5.1 Relation with ROS2 21
5.5.2 Elements of the Simulation 22

6 Conclusion **23**

List of Figures

1 Count of Each Class 9
2 System Architecture 11
3 Reported Scores 13
4 Example of Predictions 15
5 System Architecture: Node Communication and Hardware Hierarchy 21
6 Drone model integrated with the necessary sensors 22
7 The agricultural farm image 22
8 A simulated tomato plant 22

Abstract

n Africa, crop diseases and pests cause significant yield losses, reducing production by up to 40% annually, which threatens both food security and the region's economy. Agriculture, employing over 60% of the population and contributing about 23% to the region's GDP, is particularly vulnerable to diseases such as tomato leaf curl virus and pepper blight. These challenges are further exacerbated by climate change and limited access to advanced agricultural technologies.

This project aims to address the growing threat of crop diseases by developing robust machine learning models capable of detecting and identifying diseases in images of key crops, including corn, pepper, and tomatoes. The models must not only accurately predict known diseases but also generalize to unseen diseases, ensuring reliable performance. Additionally, the models are optimized for deployment on drones, widely used by subsistence farmers in Africa.

By leveraging machine learning, this project seeks to enhance disease detection, improve crop productivity, and contribute to food security for millions, ultimately fostering agricultural sustainability in the region.

1 Overview and Objectives

1.1 Overview

This project integrates drone technology with artificial intelligence (AI) to detect diseases in tomato and pepper crops. The primary goal is to enhance agricultural practices by providing early, accurate detection of diseases through high-resolution drone imagery, processed using advanced AI algorithms. By leveraging drone-based data collection and AI-based analysis, we aim to assist farmers in identifying crop diseases early, ultimately contributing to increased yields and food security.

1.2 Objectives

The main objectives of the project are:

- To develop a scalable solution for early disease detection in tomato and pepper crops using drone technology.
- To implement AI algorithms that can accurately classify various crop diseases.
- To ensure that the solution is affordable and deployable in remote areas with minimal infrastructure.
- To improve community resilience and reduce crop loss by providing actionable insights for disease management.

2 Solution Description and Purpose

2.1 Solution Description

The project is divided into two parts:

- **Drone Part:** Drones are used to collect aerial imagery of tomato and pepper crops. These drones are equipped with high-resolution cameras that capture detailed images of the crops, even from large areas. The drones can cover fields quickly, providing real-time updates on crop health.

- **AI Part:** Machine learning models, specifically convolutional neural networks (CNNs), analyze the collected images to detect and classify crop diseases. These models are trained to recognize symptoms of common crop diseases, including both known and potentially unseen diseases.

2.2 Purpose

The purpose of this solution is to create a reliable, real-time system that supports farmers in managing crop diseases more effectively. This system will minimize crop losses by detecting diseases early and suggesting interventions before major damage occurs.

3 Objectives and Expected Outcomes

3.1 Algorithm Development

The AI models developed will focus on identifying diseases based on drone imagery. The key steps involved include:

- Image preprocessing to enhance the quality of drone-captured images.
- Training convolutional neural networks (CNNs) to classify diseases in tomatoes and peppers.
- Ensuring that the models can generalize to unseen diseases, improving robustness.
- Optimizing the models for deployment on drones and edge devices for real-time analysis.

Expected outcomes:

- A highly accurate AI model capable of identifying diseases in drone-captured images.
- Efficient, real-time processing of drone data on edge devices.

3.2 Enhanced Disaster Response

By detecting diseases early, the system will allow for rapid responses to prevent crop loss during outbreaks. Drones can regularly monitor crops and alert farmers when a disease is detected, enabling quick interventions. This will enhance disaster response by providing timely data that prevents large-scale crop failures.

Expected outcomes:

- Increased ability to mitigate the impact of crop diseases and pest infestations.
- Faster disaster response times, reducing the overall damage to crops.

3.3 Community Resilience

The project will contribute to building resilience in farming communities by reducing their vulnerability to crop diseases. By empowering farmers with timely and accurate information, they can take action to protect their crops, thereby ensuring better yields and financial stability.

Expected outcomes:

- Strengthened community resilience through the adoption of technology-driven agriculture.
- Improved food security and economic stability for farmers by reducing losses due to disease outbreaks.

4 AI Part

4.1 Dataset

4.1.1 Data Sources

The image dataset used for this project was sourced from farms in Ghana, containing images of both pepper and tomato crops. These images were collected using high-resolution cameras mounted on drones, providing a comprehensive overview of the crop fields. The goal is to detect and classify diseases affecting these crops by analyzing the images using machine learning models.

4.1.2 Data Formats

The dataset consists of images stored in standard image formats such as `.jpg`. These formats are commonly used in computer vision tasks due to their compatibility with image processing libraries and machine learning frameworks.

4.1.3 Dataset Overview

The dataset comprises 3,098 images and contains 25,788 labeled objects. Each object is either a tomato or pepper plant, and all objects are classified into one of five disease categories per crop type. Below is a summary of the dataset structure:

- **Number of Images:** 3,098
- **Total Labeled Objects:** 25,788
- **Crop Types:** Tomato, Pepper
- **Disease Categories:** Each crop type has five different diseases to be detected and classified.

4.1.4 Class Distribution

The distribution of labeled objects across different disease categories for both tomato and pepper is as follows:

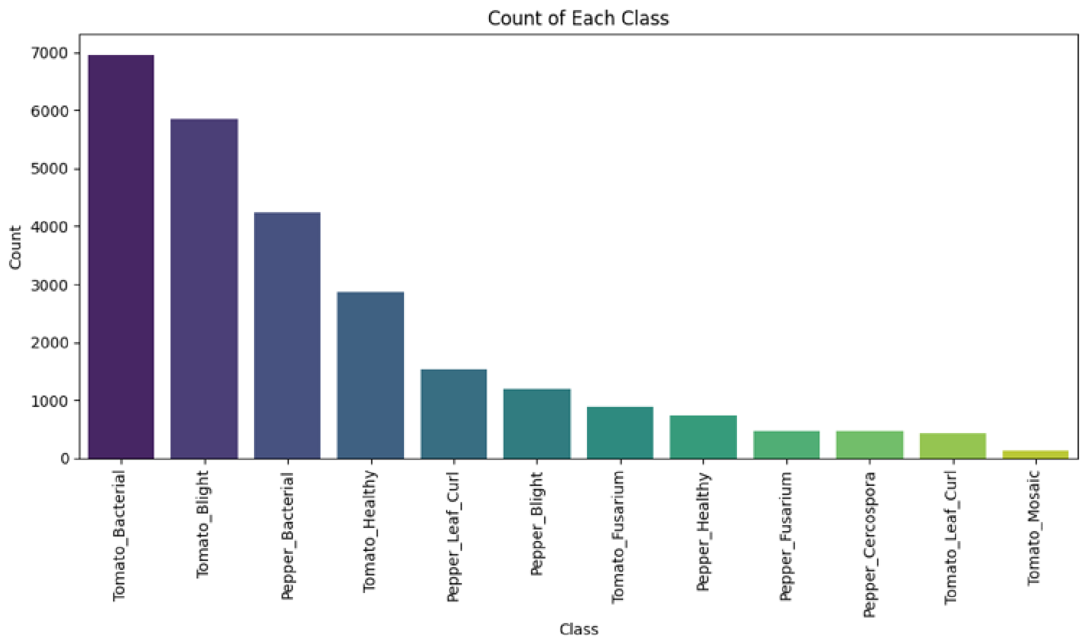

Figure 1: Count of Each Class

This dataset provides a solid foundation for training machine learning models to detect and classify diseases across the two crop types. The diversity in disease types will allow the models to generalize well and ensure accurate disease detection, even in varying environmental conditions.

4.2 Data Modeling

4.2.1 Description of Data Models

In this project, we utilize the **YOLOv11** model, a state-of-the-art object detection algorithm designed for real-time detection tasks. YOLOv11 is employed to detect and localize diseased areas in images of tomato and pepper crops. The model draws bounding boxes around the affected regions and classifies the detected diseases into one of the predefined categories for each crop. This ensures that both localization and classification are performed simultaneously, making it an efficient solution for disease detection in agricultural contexts.

4.2.2 Model

The YOLOv11 model is configured to:

- Detect diseases in tomato and pepper crops from images.
- Localize diseased areas by predicting bounding boxes.
- Classify the disease type based on the crop and specific symptoms.

The model is designed to operate efficiently on edge devices like drones, which subsistence farmers can use for disease monitoring in real-time. This allows for timely interventions to mitigate disease spread and crop loss.

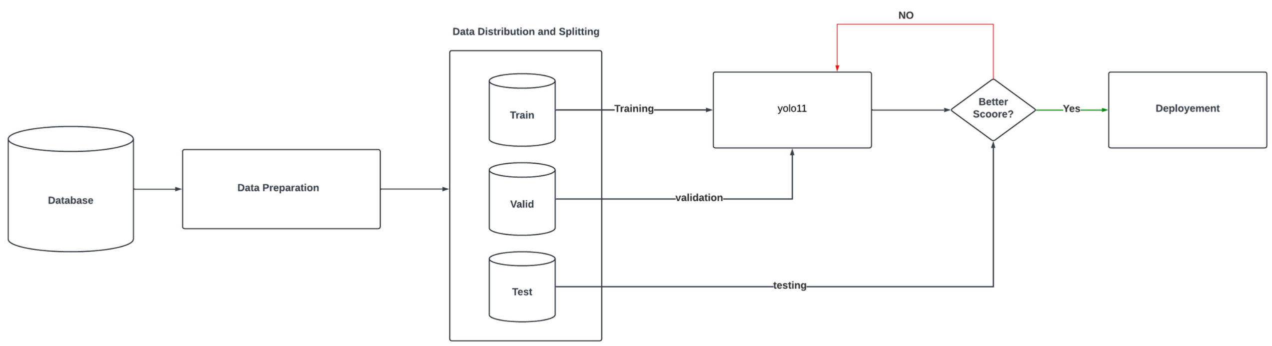

Figure 2: System Architecture

4.2.3 Model Training

The model is trained using **100 epochs** and an image size of **2048x2048** pixels to capture high-resolution details in the crops. A batch size of 8 is used during training, and the system leverages multi-threading with 7 workers to optimize data loading speed. To prevent overfitting, early stopping is employed, halting the training if no improvement in validation loss is observed for 15 consecutive epochs.

Key augmentations include:

- **Mosaic Augmentation:** Merges four images into one with a probability of 0.5, allowing the model to learn from varied contexts.
- **Horizontal and Vertical Flips:** Applied with a 50% probability, enabling the model to generalize across different image orientations.
- **Rotation and Scaling:** Random rotations up to 30 degrees and scaling variations help the model learn robust features regardless of image distortions.

4.2.4 Model Validation and Testing

During the validation process, 5-fold Group K-Fold Cross-Validation is used to split the data based on the `Image_ID`, ensuring that data leakage is avoided. The validation loss is monitored to guide the selection of the best model checkpoint. Additionally, the Intersection Over Union (IoU) threshold is set at 0.6, and the confidence threshold at 0.5 to balance between precise localization and accurate disease classification.

The trained model is tested on a held-out dataset to evaluate its performance, ensuring it can generalize well to unseen data and new disease types.

4.3 Performance Metrics

4.3.1 Model Accuracy

To evaluate the performance of the YOLOv11 model in detecting and classifying crop diseases, several key metrics are used:

- **Precision:** Precision measures the proportion of correctly predicted positive instances (diseased regions) out of all instances predicted as positive by the model. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

  A high precision score indicates that the model makes few false positive errors.

- **Recall:** Recall measures the proportion of actual positive instances (true diseased regions) that the model successfully identifies. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

  A high recall score indicates that the model can identify most of the actual diseased regions.

- **F1-Score:** The F1-score provides a harmonic mean of precision and recall, balancing the two metrics, and is useful when you need to account for both false positives and false negatives. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Intersection Over Union (IoU):** IoU is a key metric for object detection tasks. It measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as the ratio of the area of overlap to the area of union between the predicted and actual boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

  An IoU threshold of 0.6 is used, meaning that the model considers a prediction correct if the IoU between the predicted and actual bounding boxes is greater than or equal to 0.6.

- **Mean Average Precision (mAP):** This metric is used to summarize the precision-recall curve for object detection tasks. The mAP is computed as the average of the maximum precisions at different recall thresholds for each class. A higher mAP score indicates better model performance across all classes.

4.3.2 Reported Scores

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	620	4400	0.754	0.147	0.444	0.211
Pepper_Bacterial	122	837	1	0.00478	0.502	0.176
Pepper_Blight	50	236	0.5	0.0805	0.278	0.108
Pepper_Cercospora	28	68	1	0.0294	0.515	0.234
Pepper_Fusarium	44	106	0.886	0.292	0.573	0.251
Pepper_Leaf_Curl	114	316	0.673	0.117	0.379	0.146
Tomato_Bacterial	116	1398	0.71	0.0858	0.389	0.197
Tomato_Blight	115	1131	0.686	0.217	0.452	0.246
Tomato_Fusarium	23	212	0.48	0.17	0.306	0.174
Tomato_Leaf_Curl	13	84	0.605	0.31	0.458	0.26
Tomato_Mosaic	3	12	1	0.167	0.583	0.319

Figure 3: Reported Scores

These results demonstrate the model's ability to accurately detect and classify diseased regions in crop images while maintaining a balance between precision and recall. The high mAP score reflects strong performance across both tomato and pepper disease classes.

4.4 Inference

4.4.1 Input and Output of New Data

4.4.2 Input Data

The input data for the inference process consists of images captured by drones in the field. These images include various instances of tomato and pepper crops, showcasing a range of conditions and potential diseases. The model is designed to accept standard image formats such as JPG. Each input image undergoes preprocessing to ensure that it meets the necessary requirements for the YOLOv11 model, including resizing and normalization.

4.4.3 Output Interpretation

The output of the model during inference includes the following:

- **Bounding Boxes:** The model generates bounding boxes around detected diseased areas in the input images, indicating the regions of interest.
- **Class Labels:** Each bounding box is associated with a predicted class label, identifying the specific disease (if any) present in that region. The classes include different disease types for tomato and pepper crops.
- **Confidence Scores:** Along with the predicted labels, the model provides confidence scores, indicating the likelihood that the detected region corresponds to the predicted disease class. Higher scores reflect greater confidence in the model's prediction.

4.5 Example of Predictions from the Model

To illustrate the model's output, consider the following example:

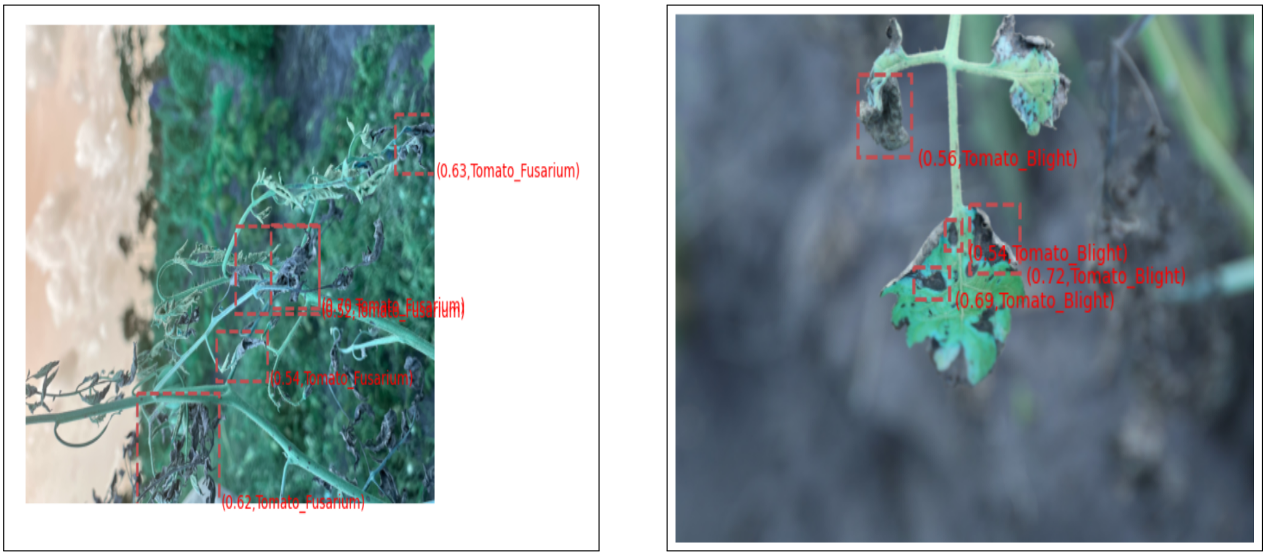

Figure 4: Example of Predictions

5 Hardware Architecture for Autonomous Drones

5.1 Introduction

This section focuses on the hardware architecture and integration of a smart autonomous drone. We will discuss the system's components, the role of ROS2 (Robot Operating System 2) in managing these components, and the use of Gazebo for simulation and testing.

5.1.1 Overview of the System

The autonomous drone consists of key hardware components such as GPS, IMU, Lidar, and an onboard computer. These elements work together to gather data, process information, and perform navigation tasks, enabling the drone to operate effectively in various environments.

5.1.2 Role of ROS2

ROS2 is the software framework that coordinates communication between different nodes managing sensors and control commands. Its modular design allows for flexibility and scalability, ensuring that the drone can adapt to changing operational needs.

5.1.3 Gazebo for Simulation

Gazebo is essential for simulating the drone's environment, allowing for thorough testing of its functionalities before real-world deployment. This simulation helps identify potential issues, ensuring the drone's reliability and safety during operation.

5.2 Hardware Components

The hardware architecture of the autonomous drone comprises several critical components that work together to facilitate its operation. Key components include:

- **Raspberry Pi 4**: The onboard computer for the drone, the Raspberry Pi 4 offers sufficient processing power and connectivity options for running ROS2 and handling sensor data. Its compact size and versatility make it an ideal choice for autonomous applications.

- **GPS Module**: The GPS module provides accurate positioning data, enabling the drone to navigate and follow pre-defined paths. This data is essential for tasks such as waypoint navigation and geofencing.

- **Inertial Measurement Unit (IMU)**: The IMU combines accelerometers and gyroscopes to measure the drone's orientation and motion. This information is crucial for maintaining stability and control during flight.

- **Lidar Sensor**: The Lidar sensor emits laser pulses to measure distances and create a detailed map of the surroundings. It aids in obstacle detection and avoidance, enhancing the drone's autonomous capabilities.

- **Camera Module**: The onboard camera provides visual data for tasks such as object detection and monitoring. It works in conjunction with AI models to identify and classify objects in real time.

- **Motor Controllers and ESCs**: Electronic Speed Controllers (ESCs) regulate the speed of the drone's motors, enabling precise control of flight dynamics. They work in tandem with the flight control algorithms to achieve stable flight.

- **Battery and Power Management System**: A reliable battery provides the necessary power for all components, while a power management system ensures efficient distribution of power to the various subsystems.

These components collectively enable the autonomous drone to gather data, process information, and navigate efficiently in diverse environments.

5.3 ROS2 Architecture

The Robot Operating System 2 (ROS2) serves as the software backbone for the autonomous drone, providing a flexible framework for managing various hardware components and facilitating communication between them. The architecture of ROS2 is based on a publisher-subscriber model, which allows different nodes to exchange information efficiently.

- **Nodes**: Each functional component of the drone is implemented as a separate node. For instance, there are nodes for handling sensor data from the GPS, IMU, Lidar, and camera. These nodes can operate independently and communicate with each other through defined topics.

- **Topics**: Topics are communication channels that nodes use to publish and subscribe to messages. For example, the Lidar node publishes distance measurements to a topic, which can be subscribed to by the obstacle detection node to identify potential hazards.

- **Services and Actions**: In addition to topics, ROS2 supports services and actions for request-response interactions. Services are used for synchronous calls, while actions enable long-running tasks, such as navigating to a specific waypoint. This structure allows for more complex interactions between nodes.

- **Quality of Service (QoS) Settings**: ROS2 provides configurable QoS settings that ensure reliable data transmission according to the application's needs. This is particularly important for real-time applications where low latency and high reliability are crucial.

- **Middleware**: ROS2 utilizes DDS (Data Distribution Service) as its underlying middleware, which enhances communication between nodes and ensures scalability. DDS facilitates the distribution of messages across different network configurations, making ROS2 suitable for a wide range of robotic applications.

Overall, the ROS2 architecture enables seamless integration of various hardware components and facilitates efficient communication, essential for the autonomous operation of the drone.

5.4 ROS2 Nodes: Inputs, Outputs, and Roles

The ROS2 system for the autonomous drone is composed of several interconnected nodes, each responsible for processing specific sensor data or performing a particular task. Understanding the input, output, and role of each node is critical to ensure smooth communication and functionality across the system.

5.4.1 GPS Node

Input: Raw GPS signals from the GPS module.
Output: Processed position data (latitude, longitude, and altitude) published to the `/gps_fix` topic.
Role: The GPS node is responsible for providing real-time global positioning information. This data is crucial for the drone's localization and is used in conjunction with other sensors to determine the exact position and trajectory of the drone during navigation tasks.

5.4.2 IMU Node

Input: Inertial data from the IMU (accelerometer, gyroscope readings).
Output: Orientation and angular velocity data published to the `/imu_data` topic.
Role: The IMU node tracks the drone's orientation and motion, which is essential for stabilizing the drone during flight. This data, when fused with GPS information, ensures precise control over the drone's position and trajectory.

5.4.3 Lidar Node

Input: Distance measurements from the Lidar sensor.
Output: A point cloud representing the surrounding environment, published to

the /scan topic.
Role: The Lidar node detects obstacles and generates a real-time map of the drone's environment. This information is used by the path-planning and obstacle avoidance nodes to ensure safe navigation.

5.4.4 YOLO11 Node (AI-based Crop Disease Detection)

Input: Images captured by the onboard camera.
Output: Processed images with detected signs of crop diseases and bounding boxes, along with confidence scores, published to the /detected_diseases topic.
Role: The camera node uses a YOLOv11 model for real-time detection of crop diseases. The node identifies affected areas in the drone's field of view, classifies the type of disease, and determines its location, aiding in precise monitoring and treatment of crops.

5.4.5 Path Planning Node

Input: Target destination (from a mission planner or user input) and sensor data from the Lidar and GPS nodes.
Output: A dynamically calculated path, published to the /planned_path topic.
Role: This node computes an optimal path for the drone based on the target location and sensor data. It avoids obstacles by adjusting the path in real time, ensuring efficient and safe navigation to the goal.

5.4.6 Navigation Node

Input: The planned path from the Path Planning node and real-time position data from the GPS and IMU nodes.
Output: Velocity commands and motor control signals published to the /cmd_vel topic, controlling the drone's movement.
Role: The navigation node is responsible for following the planned path and issuing commands to adjust the drone's velocity and orientation. It ensures that the drone reaches its destination accurately while avoiding obstacles.

Each node operates independently but is interconnected through ROS2 topics, enabling a modular, scalable system where real-time data is processed and acted upon seamlessly. This architecture ensures that the drone can perform complex tasks like navigating to specific waypoints, detecting and avoiding obstacles, and interacting with its environment autonomously.

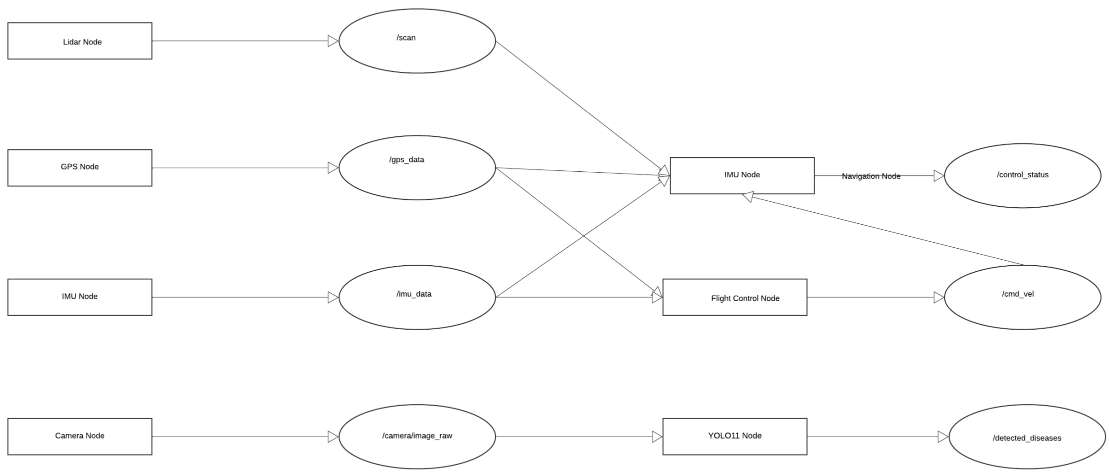

Figure 5: System Architecture: Node Communication and Hardware Hierarchy

5.5 Gazebo Simulation Overview

Gazebo is a powerful open-source robotics simulation platform that enables the development and testing of robotic systems in realistic 3D environments. It provides a flexible and intuitive interface for modeling various scenarios, making it particularly useful for simulating autonomous drones.

5.5.1 Relation with ROS2

Gazebo integrates seamlessly with the Robot Operating System 2 (ROS2), allowing for effective communication between different software components. This integration facilitates the simulation of the complete software stack used in the drone, enabling the testing of algorithms for navigation, perception, and control within a simulated environment.

5.5.2 Elements of the Simulation

In our project, the Gazebo simulation includes several key elements:

- **Drone Model:** A 3D representation of the drone equipped with various sensors, such as cameras and Lidar. The sensors are configured using SDF (Simulation Description Format) coding to accurately reflect their physical properties and behavior.
- **Agricultural Environment:** A virtual farm setting featuring crops and obstacles to simulate real-world scenarios for crop monitoring and disease detection.
- **Sensor Integration:** Implementation of simulated sensors that provide data for navigation and object detection, mirroring the hardware used in the actual drone.

This setup allows us to visualize and test the drone's operations before deploying it in real-world conditions, ensuring functionality and safety.

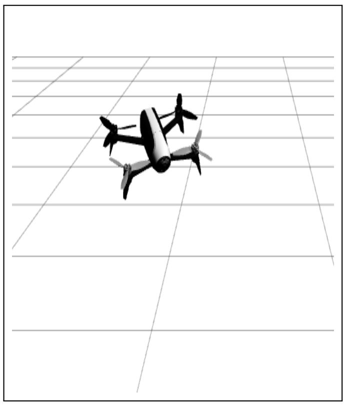

Figure 6: Drone model integrated with the necessary sensors

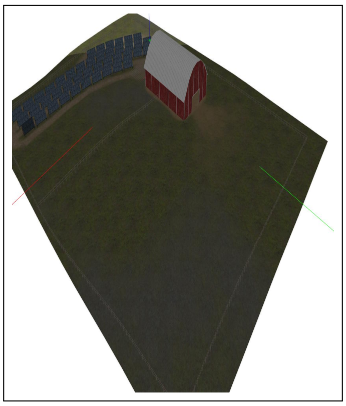

Figure 7: The agricultural farm image

Figure 8: A simulated tomato plant

6 Conclusion

This project successfully integrated AI-based disease detection with drone technology to provide an efficient and scalable solution for identifying crop diseases in tomato and pepper fields. By leveraging advanced machine learning models, we were able to detect diseases with high accuracy, while the use of drones facilitated rapid data collection over large areas, making the system practical for real-time monitoring.

The combination of AI and drones offers a powerful tool to support farmers in early disease detection, reducing crop losses and improving agricultural productivity. The system's adaptability for deployment in remote areas using low-resource devices also enhances its potential for broad application in regions with limited access to advanced agricultural technologies.

Overall, this project provides a strong foundation for future developments in smart agriculture, with the potential to greatly improve food security and promote sustainable farming practices.