Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

A framework for AI-based plant disease detection and autonomous robotic agricultural spraying

Burhan ÖK[1], Kenan IŞIK[2*]

[1] Karabuk University, Mechatronics Engineering Department, 3burhanok@gmail.com, Orcid No: 0000-0003-3511-8482
[2] Karabuk University, Mechatronics Engineering Department, kenanisik@karabuk.edu.tr, Orcid No: 0000-0002-0973-5180

ARTICLE INFO

Article history:

Received 30 December 2024
Received in revised form 11 January 2026
Accepted 16 March 2026
Available online 19 March 2026

Keywords:

agricultural spraying, artificial intelligence, mobile manipulation

Doi: 10.24012/dumf.1608369

* Corresponding author

ABSTRACT

This study presents a novel framework for plant disease detection using artificial intelligence (AI) and efficient agricultural spraying via a mobile robotic manipulator. The dataset used to train the AI model was created by capturing images of plant leaves with and without disease symptoms and labeling them according to the YOLO algorithm. A camera equipped with a depth sensor was employed to detect plant diseases, and the location of the diseased regions relative to the end effector was computed using kinematic methods. The Robot Operating System (ROS) was used for system integration, while the MoveIt! package facilitated kinematic calculations and motion planning of the robotic arm. The robotic arm was mounted on a two-wheeled mobile platform that autonomously navigates among plants using the ROS Navigation Stack. By enabling autonomous robotic spot spraying of diseased areas, the proposed system reduces both labor requirements and pesticide usage in agricultural spraying, thereby lowering pesticide costs and minimizing consumer exposure. The framework proposed in this study represents a unique application in which disease detection is integrated with autonomous robotic agricultural spraying at the spot-spraying level.

Introduction

The rapid increase in the global population is making the production and supply of food—our most basic necessity—more important than ever. However, the agricultural sector faces various challenges, including migration from rural areas to cities [1], the resulting shortage of agricultural labor, high production costs due to fertilizer use, and seasonal droughts. These challenges make efforts to improve efficiency in agricultural production increasingly critical [2].

The most common and traditional practices used to increase agricultural productivity are fertilization and pest control. Fertilization supplies plants with the nutrients required for healthy and rapid growth, while pest control prevents the development and spread of plant diseases that negatively affect growth and reduce production efficiency. Although fertilization can be applied to a desired area in controlled amounts, broadcast spraying is often preferred in pest control due to its simplicity. In this method, either the entire plant or the entire cultivated area is sprayed. While tractors and airplanes have traditionally been used for spraying, drones and mobile robots have also begun to be adopted in recent years [3].

Despite its ease of application, the broadcast spraying method has several significant disadvantages. The most notable drawback is that the entire plant is sprayed rather than only the diseased parts. This practice can result in agricultural chemicals reaching consumers through plants or their fruits, thereby posing risks to human health. Reducing pesticide use in agriculture positively affects public health and has been a major objective of the European Union; Directive 2009/128/EC explicitly aims to reduce the risks and impacts of pesticide use on human health and the environment, and the EU's Farm to Fork Strategy targets a 50% reduction in the use and risk of chemical pesticides by 2030 [4]. Another disadvantage of broadcast spraying is the excessive amount of chemicals used, which significantly increases spraying costs.

Considering both spraying costs and the high level of consumer exposure to agricultural chemicals, spot spraying emerges as a more efficient alternative to broadcast spraying. However, spot spraying is not widely adopted due to longer operation times, the lack of skilled agricultural labor, and high overall labor costs.

The use of robotic systems is one of the most effective approaches for reducing labor costs across many industries. Robotics enables improved continuity,

standardization, and production efficiency. Artificial intelligence (AI), in turn, enhances robotic systems by enabling decision-making based on learning and knowledge, and has been widely recognized for its success in tasks requiring expert-level performance. AI plays a significant role in diagnosing, identifying, and selecting treatment methods in medicine and many other disciplines [5].

In the literature, agricultural spraying studies primarily focus on tractors, drones, and mobile robots [6], each of which offers distinct advantages and limitations. Tractors have long played a significant role in agricultural spraying and remain the primary solution for covering large areas. When equipped with sensors and cameras, tractors can perform more precise spraying operations [7]. However, conventional tractor-based spraying is limited to open fields, contributes to soil compaction, and can damage plant roots—thereby reducing productivity [8], [9]—and is generally not suitable for greenhouse environments. Furthermore, broad-area spraying with tractors can lead to excessive chemical use compared to more targeted approaches [10], [11].

In recent years, drones have gained popularity in agricultural spraying due to their ability to cover large areas quickly, their lower operational costs compared to tractors, and their autonomous capabilities [6], [12]. Nevertheless, similar to tractors, drones also tend to use large amounts of chemicals and are unsuitable for greenhouse environments, which limits their applicability. Another spraying approach involves the use of mobile robots. Agricultural mobile robots are typically designed to operate either over or among plants. Although their relatively low operating speed makes them unsuitable for spraying large areas, mobile robots excel in precision spraying, adaptability to both open-field and greenhouse environments, and seamless integration with AI and image-processing techniques [13]–[15]. However, drawbacks include the potential for excessive chemical use and the need for customized designs depending on plant height.

In this study, a framework for spraying diseased plants is developed by integrating a mobile robot with a robotic arm. The inclusion of a robotic arm enables more precise and selective spraying [16]. The proposed system's autonomous navigation capability, AI-based disease detection, selective spot-spraying functionality, greenhouse operability, and adaptability to tasks beyond spraying enhance its versatility. Since targeted spraying relies on AI-driven image processing, the quality of the visual dataset used for disease detection is critical. A review of available open-source datasets revealed that many focus solely on individual leaves. As such datasets were unsuitable for this work, a new visual dataset covering the entire height of the plant was created, thereby contributing to the literature on agricultural datasets. Although the platform can operate in both open-field and greenhouse environments, it is particularly well suited for greenhouses, where environmental factors such as lighting, humidity, and dust are controlled. For open-field deployment, additional real-world testing and calibration are required.

Existing studies in agricultural robotics typically focus on environmental mapping [17], AI-based plant or disease recognition [18]–[24], and spraying operations using robotic arms [16], [25]. In many cases, mobile robotic spraying targets the entire plant [13], [14], [26], which reduces spraying efficiency and increases pesticide consumption. To address this issue, Oberti et al. implemented spot spraying for powdery mildew in grapevines using a six-degree-of-freedom robotic manipulator mounted on a mobile base [16]. Building on this work, the present study proposes a framework based on open-source software tools for spot spraying three plant diseases: chlorosis, powdery mildew, and septoria. First, plant diseases are detected using an AI model trained on a custom visual dataset. Subsequently, a depth camera is used to determine the positions and orientations of diseased areas. By guiding the robotic arm to the appropriate locations for spot spraying, the proposed approach significantly reduces pesticide usage compared to broadcast spraying methods, thereby minimizing chemical exposure and mitigating potential adverse effects on human health.

Although this study focuses on disease detection and pesticide spraying, the proposed framework can be readily adapted for additional agricultural applications, such as ripeness monitoring, automated selective harvesting, and insect cluster detection and spraying.

Methodology

Preparation of Dataset

In the training of artificial intelligence (AI) models, both supervised and unsupervised learning methods are commonly used. When supervised learning is employed, both the input data and the corresponding output labels must be provided to the AI model. In this manner, the model learns which inputs correspond to which outputs, and the training process is carried out. When such data are combined, datasets are created. These datasets may contain thousands, millions, or even billions of data samples. Various types of datasets can be used to develop AI applications, including sequential, visual, numerical, two-dimensional (2D), and three-dimensional (3D) data [27].

Since the objective of this project is to detect diseases on plant leaves, a dataset composed of visual data is required. Visual datasets typically contain more data than other dataset types, making them more time-consuming to process and prepare. Visual datasets may have a 2D structure (e.g., grayscale images) or a 3D structure (e.g., RGB color images). In object detection tasks, models often rely on color information; therefore, RGB images are generally preferred [28].

Model training can be performed using either pre-existing datasets or newly created datasets. Many open-source datasets are available online, often for educational or research purposes. However, for specialized applications, suitable datasets may not always exist. In such cases, a

custom dataset tailored to the requirements of the project must be created. This can be achieved either by collecting images through web scraping or by capturing photographs directly. Since manual image acquisition through photography is a time-consuming process, it is generally considered a secondary option when suitable online data are available.

In this project, three common leaf diseases—chlorosis, powdery mildew, and septoria—were selected as the focus. These diseases were chosen because they are frequently encountered on plant leaves, facilitating dataset preparation. Nevertheless, detection algorithms can also be developed for other types of leaf diseases, fruit diseases, or insect infestations on leaves and fruits.

The dataset used in this study consists of 526 images obtained through a two-stage process. In the first stage, diseased plants were identified, and each affected leaf was photographed individually using a camera. In the second stage, publicly available plant images were collected using web scraping techniques. Of the total images, 420 were obtained through direct photography, while 106 were collected via web scraping. The dataset includes 198 images of chlorosis, 152 images of powdery mildew, and 176 images of septoria. Of the 526 images, 450 were used for training the AI model, and the remaining 76 were reserved for testing. Sample images from the dataset are shown in Figure 1.

Figure 1. Sample images from the dataset.

For object recognition and object detection tasks, it is common practice to resize images to ensure uniform dimensions. However, resizing is not strictly required for object detection when bounding box annotations are normalized. Since both object recognition and object detection tasks are performed in this project, the original image dimensions were preserved. To facilitate dataset labeling, a numerical naming convention was adopted for all images.

Dataset labeling plays a crucial role in the training of AI models. During the labeling process, the expected output corresponding to a given input is specified, enabling the model to learn how to predict results for unseen data. Different labeling approaches are used for object recognition and object detection tasks. In object recognition, the goal is to teach the model what the object is, whereas in object detection, the objective is to identify both the object class and its location within the image. Consequently, the labeling process varies depending on the task.

For object recognition, several ready-to-use software tools are available for dataset labeling. In contrast, object detection requires manual annotation, where each object within an image is enclosed in a bounding box and assigned a class label. This process can be carried out using either desktop applications or online tools [29]. In this study, labeling was performed online using the MakeSense.ai tool, following the YOLO data format:

<class id>< X_0/X>< Y_0/Y><W/X><H/Y>

where:

- class_id indicates the class index of the object,
- X_0 and Y_0 represent the x and y coordinates of the bounding box center,
- W and H denote the width and height of the bounding box,
- X and Y represent the width and height of the image, respectively.

Figure 2 illustrates these label parameters on a sample image from the dataset. During the labeling process, numerical values for each parameter are determined, and a separate text file is automatically generated for each image. Since an image may contain multiple diseased regions, a single text file may include multiple bounding box annotations.

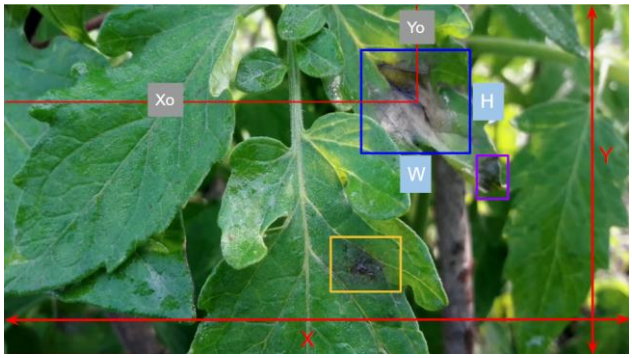

Figure 2. Visualization of the labels on an image.

Learning Algorithm and Model Training

In artificial intelligence, various approaches are used to develop learning algorithms, such as Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs). Among these approaches, CNNs are the most suitable for working with spatial data such as images. Since this project requires training a model using images and involves both object detection and object recognition tasks, CNNs represent the most appropriate approach.

CNNs, which are primarily used for classification and object detection problems, are deep learning algorithms that take an input image and apply multiple filters, represented as matrices, to extract specific features. A CNN consists of multiple layers, including an input layer, convolutional layers, pooling layers, and fully connected layers [30]. An input image is processed sequentially

through these layers to extract features. Implementing and optimizing these processes from scratch can be challenging; therefore, ready-to-use libraries are commonly employed. The most widely used libraries in this field are TensorFlow (developed by Google) and PyTorch (developed by Facebook).

Since this project involves both object recognition and object detection, using these libraries alone is insufficient, and more specialized algorithms are required. Customized detection architectures can be developed using TensorFlow or PyTorch. Examples of such algorithms include the TensorFlow Object Detection API, YOLO, Fast R-CNN, Faster R-CNN, Fastest R-CNN, and SSD, each of which has its own advantages and limitations. In this study, the YOLOv3 (You Only Look Once) algorithm was selected for training the AI model. YOLO performs one-stage object detection and localizes objects using bounding boxes. By incorporating depth information obtained from the Kinect sensor, three-dimensional localization of the diseased areas on plants is achieved.

Although the YOLO algorithm provides pre-trained models, these were not used in this project because their datasets primarily include objects such as cars, people, and everyday items, which are outside the scope of agricultural disease detection. Consequently, a custom dataset was created, as described in the Preparation of Dataset subsection, and was used for both training and testing. Since model training requires substantial GPU computational power, the training process was conducted on the Google Colab platform.

Training the AI model took approximately 10 hours and was performed over 3000 iterations. After every 1000 iterations, the model weights were automatically saved. It was observed that after 2000 iterations, the learning rate degraded significantly, and signs of overfitting appeared. Therefore, the weights obtained after 2000 iterations were selected for use in this study.

Upon completion of the training process, the required weight files were generated, rendering the model ready for integration into the proposed system.

The performance of the trained AI model was evaluated using several metrics. The mean Average Precision at an Intersection over Union threshold of 0.5 (mAP50) was used as the primary performance metric. At the end of training, the model achieved an mAP50 score of 0.7545, indicating strong performance in detecting plant diseases.

A separate test dataset consisting of 76 (50 diseased, 26 healthy plant) images was used for performance evaluation. Object detection experiments conducted on this dataset demonstrated that the model achieved an overall success rate of 83% in real-world scenarios. The details of the disease detection experiment performed directly on plant images are given in Table 1.

The confusion matrix indicates that the trained AI model correctly identified 42 out of 50 diseased samples, resulting in a true positive rate of 84%. This suggests that the model is generally effective at detecting diseased leaves under the tested conditions. However, 8 diseased samples were misclassified as healthy (false negatives), which is a non-negligible limitation, as missed detections directly affect the downstream spraying task.

For healthy samples, the model correctly classified 21 out of 26 instances, corresponding to a true negative rate of approximately 81%. The presence of 5 false positives indicates that some healthy leaves were incorrectly classified as diseased, which could lead to unnecessary spraying. While this behavior may be acceptable in a safety-oriented agricultural context where missing a disease is more critical than over-spraying, it still impacts the overall efficiency of pesticide usage.

Overall, the confusion matrix reflects a reasonable balance between sensitivity and specificity, but it also highlights that both false negatives and false positives remain sources of error.

Table 1: Confusion Matrix for the AI model's performance

	Predicted Positive	Predicted Negative	Total
Actual Positive	**42 (TP)**	8 (FN)	50
Actual Negative	5 (FP)	**21 (TN)**	26
Total	47	29	**76**

The detailed performance metrics are summarized in Table 2. Based on the confusion matrix, the trained AI model achieved a precision of 89.4%, indicating a low false-positive rate in disease detection. The recall was 84.0%, showing that the majority of diseased samples were correctly identified. The resulting F1-score of 86.6% reflects a balanced performance between precision and recall, demonstrating the effectiveness of the model for plant disease detection within the tested dataset.

Table 2: AI Model Performance Metrics

Metric	**Value**
mAP50	0.7545
Intersection over union (IoU)	> 0.5
Precision (TP / (TP + FP))	~0.894
Recall (TP / (TP + FN))	0.84
F1-score (2·Precision·Recall) / (Precision + Recall)	~0.866

In conclusion, the AI model developed using the YOLO algorithm demonstrates the capability to detect plant diseases with high accuracy. This achievement represents an important step toward automating disease detection in agriculture and enabling early intervention. With the dataset preparation and model training processes completed and the final weight files obtained, the next stage involves integrating the trained model into the robotic control framework.

Preparation of the Simulation Environment and the Mobile Manipulator

Agricultural spraying using an autonomous mobile manipulator is a complex process consisting of multiple stages. These stages include planning and controlling a trajectory that enables autonomous movement of the mobile platform within its working environment, scanning plants during autonomous navigation to detect diseased areas, and controlling the robotic arm to accurately spray the detected diseased regions. ROS was used to integrate these simultaneous and independent processes effectively. ROS is a collection of software libraries and tools designed to simplify the integration of hardware and software components in robotic systems. In this project, ROS packages were used for localization of the mobile base, design and mapping of the working environment, control of the manipulator, and visualization of the overall disease detection and agricultural spraying process.

To evaluate the performance of the developed AI model and the designed mobile manipulation system in a virtual environment, a workspace containing plant models was created, allowing the mobile manipulator to move freely. Several ROS packages are available for visualizing simulated robotic environments, among which Gazebo is one of the most widely used. In this study, the Gazebo simulation platform was employed to create the simulation environment.

For disease detection, plant models containing diseased leaves were required in the simulation environment. To achieve this, open-source plant models were downloaded from online 3D model repositories, and disease symptoms were integrated into the leaf surfaces using textures derived from real images in the dataset and applied using Blender. The finalized plant models were then imported into ROS and Gazebo in .dae file format.

Since the implementation of the framework was conducted entirely in a simulated environment, sensor tests under real environmental conditions such as dust, humidity, and lighting were not performed in a physical setting. Instead, these factors were partially emulated in the simulation environment by incorporating simple noise models.

In the ROS environment, the physical structure of robots is commonly defined using the Universal Robot Description Format (URDF). Using the URDF package, not only the robot model itself but also the attached sensors, actuators, and environmental interactions can be described. Figure 3 illustrates the structure of the mobile base and the robotic arm along with the integrated sensors and peripherals.

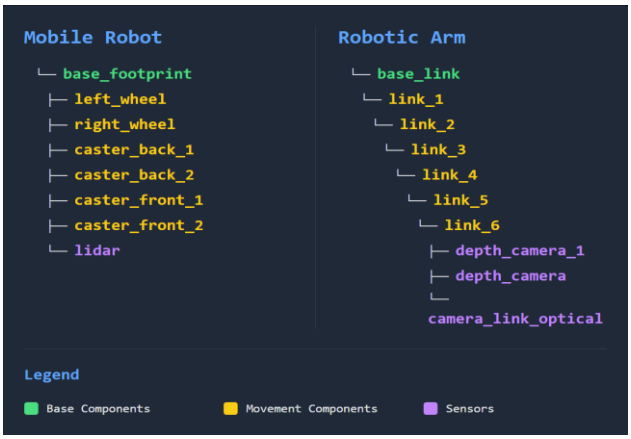

Figure 3. Visual representation of the structure of the mobile base and the robotic arm.

To detect and localize diseased areas on plants, the mobile manipulator must autonomously traverse the environment. This task requires simultaneous environment mapping and path planning. For mapping and obstacle detection, a LiDAR sensor was mounted on the mobile base, and the gmapping package was utilized within ROS. Path planning was performed using the ROS Navigation Stack.

For trajectory planning and kinematic calculations enabling the mobile manipulator to autonomously navigate the workspace and spray the diseased areas detected by the AI model, the MoveIt! package—one of the standard motion-planning libraries in ROS—was employed.

Finally, the prepared mobile manipulator model was placed into the designed simulation workspace. Figure 4 presents the mobile manipulator model configured for the planned operational scenario.

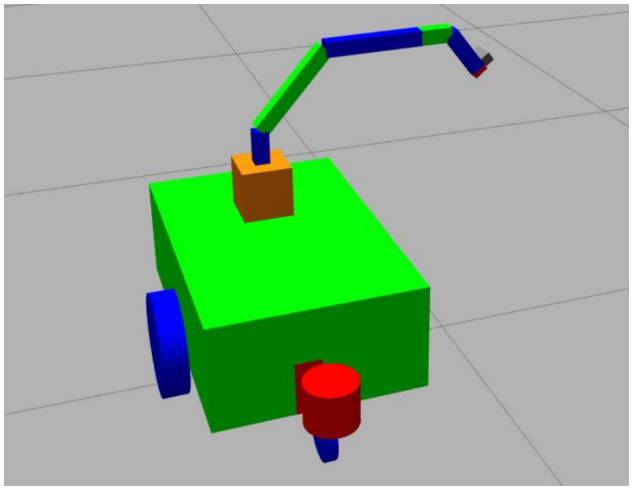

Figure 4. Visualization of the mobile manipulator in the Gazebo environment.

Disease Detection and Localization

The disease detection and localization process begins with the autonomous navigation of the mobile base within the field. To create a field model, ready-to-use shapes available in Gazebo were used, and multiple plant models with both healthy and diseased parts were added. Figure 5 shows the prepared field model.

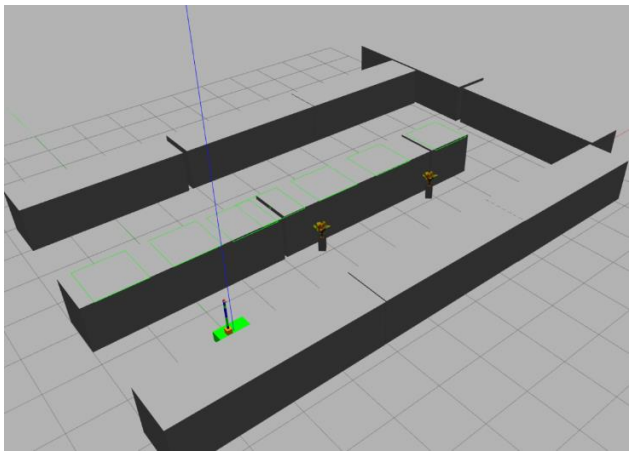

Figure 5. Field model in the Gazebo environment.

To enable the robot to perceive its surroundings, the mobile manipulator must first map the field. In this project, field mapping is performed using the gmapping package in ROS. The gmapping package utilizes point cloud data obtained from the LiDAR sensor mounted on the mobile base. Figure 6 illustrates the mapping process of the mobile base in the Gazebo environment.

Once the mapping process is complete, the mobile manipulator traverses the field to detect diseased areas on the plants and perform agricultural spraying. During this operation, it is essential to generate a safe path for the mobile base to avoid collisions with objects in the field. In this project, the ROS navigation stack package is used for path planning and motion control of the mobile base.

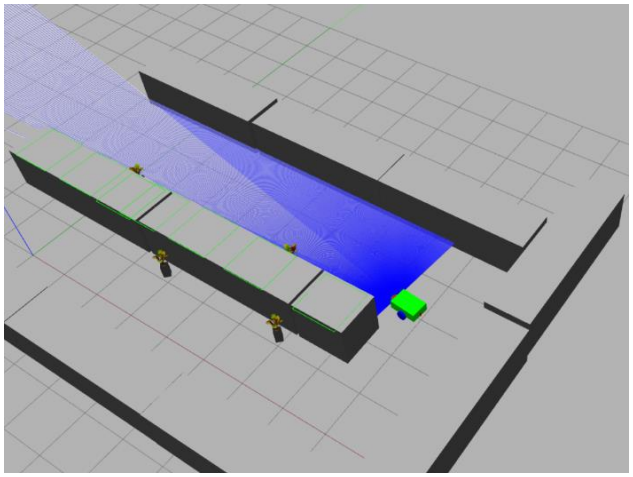

Figure 6. Visualization of the mapping process.

Given a target point, the navigation stack uses the previously generated map of the field to compute a safe trajectory for the mobile base by employing both global and local path planning methods. During navigation, the Monte Carlo Localization (MCL) algorithm is used to estimate and continuously update the robot's pose within the environment. Figure 7 presents a visualization of the path planning procedure.

While moving along the planned trajectory, the mobile base stops in front of each plant and scans it for diseases using a 3D camera. In this project, a Microsoft Kinect sensor is used, which provides high-resolution RGB images as well as depth information in the form of point cloud data. This depth information is critical for accurately determining the spatial location of the diseased areas relative to the mobile base. The visual data are transferred to OpenCV and processed using the YOLOv3 algorithm.

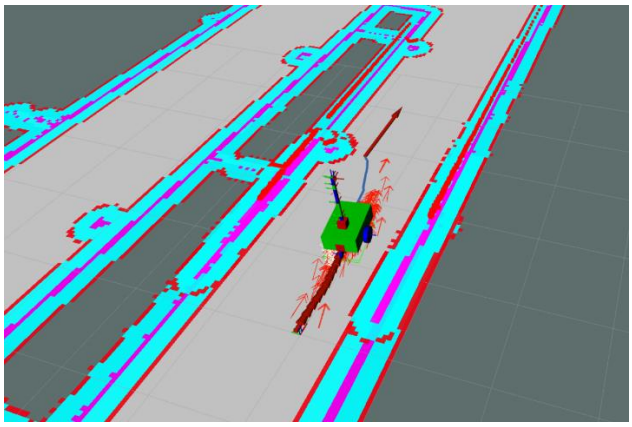

Figure 7. Visualization of the path planning process in the RViz environment.

Using the trained model, the YOLO algorithm detects diseased regions in the visual data and outputs their locations within the image. By combining the image-based detection results with the depth information provided by the 3D camera, the three-dimensional coordinates of the diseased areas are calculated.

Agricultural Spraying Using Robot Manipulator

The final step of the overall process is to apply pesticide using a nozzle as the end effector. Using the location coordinates of the diseased area, a target position and orientation for the end effector must be calculated. For effective spraying, the pesticide must be applied directly to the diseased area at a 90° angle. Therefore, not only the coordinates of the target point but also the orientation of the surface must be defined.

To define a surface in 3D space, at least three non-collinear points are required. These three points are generated by defining a small circle centered at the diseased area coordinates obtained from the YOLO algorithm. If the image coordinates of the diseased area are defined as

$$p_{d_{im}} = (x_{d_{im}}, y_{d_{im}}),$$

three points on this circle, separated by 120° from each other, can be defined as

$$p_{1_{im}} = (x_d + r\,cos\,(0), y_d + r\,sin(0)) \quad (1)$$

$$p_{2_{im}} = (x_d + r\,cos\left(\frac{2\pi}{3}\right), y_d + r\,sin\left(\frac{2\pi}{3}\right)) \quad (2)$$

$$p_{3_{im}} = (x_d + r\,cos\left(\frac{4\pi}{3}\right), y_d + r\,sin\left(\frac{4\pi}{3}\right)) \quad (3)$$

where $r$is the radius of the circle in pixels.

By projecting these points onto the point cloud data and selecting the closest points in the cloud, three corresponding 3D points are obtained. Denoting these

points in the camera frame as ${}^{c}p_1, {}^{c}p_2$ and ${}^{c}p_3$, the positions of these points in the global coordinate system can be calculated as

$$p_1 = {}^{0}_{c}T \, {}^{c}p_1 \tag{4}$$

$$p_2 = {}^{0}_{c}T \, {}^{c}p_2 \tag{5}$$

$$p_3 = {}^{0}_{c}T \, {}^{c}p_3 \tag{6}$$

where ${}^{0}_{c}T$ is the transformation matrix of the camera coordinate system relative to the global coordinate system.

A surface can be represented by its normal vector, which can be computed using two vectors lying on the surface. These vectors are defined as

$$\vec{v}_1 = p_2 - p_1 \tag{7}$$

$$\vec{v}_2 = p_3 - p_1 \tag{8}$$

The cross product of these vectors yields the surface normal

$$\vec{n} = \vec{v}_1 \times \vec{v}_2 \tag{9}$$

The normal vector always points toward the camera based on the chosen vector order. Since its magnitude depends on the lengths of $\vec{v}_1$and $\vec{v}_2$, the vector must be normalized before use:

$$\vec{n}_n = \frac{\vec{n}}{\| \vec{n} \|} \tag{10}$$

For effective spraying, the nozzle must be oriented toward the surface and positioned at a predefined distance from it. The target point for the nozzle is obtained by placing the normalized surface normal at the disease location, which is determined by projecting $p_{d_{im}}$onto the point cloud and selecting the closest point. Scaling this vector by a predefined distance yields the target position:

$$\vec{p}_t = {}^{0}_{c}T \, {}^{c}p_d + s \, \vec{n}_n \tag{11}$$

where $\vec{p}_t$is the target position of the spraying nozzle, ${}^{c}p_d$is the disease location relative to the camera, and $s$is a scaling factor that determines the desired nozzle distance from the surface.

Assuming that the z-axis of the coordinate system attached to the nozzle points in the spraying direction, the desired z-axis orientation can be defined by inverting the surface normal:

$$\vec{z}_t = -\vec{n}_n \tag{12}$$

where $\vec{z}_t$represents the z-axis of the nozzle orientation.

Rotation around the nozzle's z-axis does not affect spraying performance; therefore, any orientation satisfying this constraint is acceptable. However, to prevent the end effector and camera from rotating upside down, the y-axis of the target coordinate system is constrained to lie on a vertical plane together with the z-axis. To achieve this, an "up" direction vector is first defined. Assuming the z-axis of the world coordinate system points upward, the up vector in the camera frame is obtained as

$$\vec{z}_{up} = {}^{0}_{c}R^{T} \, [0 \; 0 \; 1]^{T} \tag{13}$$

where ${}^{0}_{c}R$ is the rotation matrix of the camera relative to the world coordinate system.

The cross product of $\vec{z}_{up}$and $\vec{z}_t$produces a horizontal vector:

$$\vec{x}_h = \vec{z}_{up} \times \vec{z}_t \tag{14}$$

Since the angle between these vectors is unknown, $\vec{x}_h$ is normalized to obtain the x-axis of the target coordinate system:

$$\vec{x}_t = \frac{\vec{x}_h}{\| \vec{x}_h \|} \tag{15}$$

Finally, the y-axis of the target coordinate system is obtained by

$$\vec{y}_t = \vec{z}_t \times \vec{x}_t \tag{16}$$

The vectors $\vec{x}_t$, $\vec{y}_t$, and $\vec{z}_t$together define the rotation matrix of the target coordinate system. The homogeneous transformation matrix of the target frame relative to the world frame is therefore expressed as

$${}^{0}_{t}T = \begin{bmatrix} \vec{x}_t & \vec{y}_t & \vec{z}_t & \vec{p}_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

This transformation matrix is then provided to the MoveIt framework to perform inverse kinematics and compute the required joint angles for positioning the nozzle at the target location and orientation to perform agricultural spraying. Figure 8 illustrates the overall process, from plant image acquisition to physical spraying.

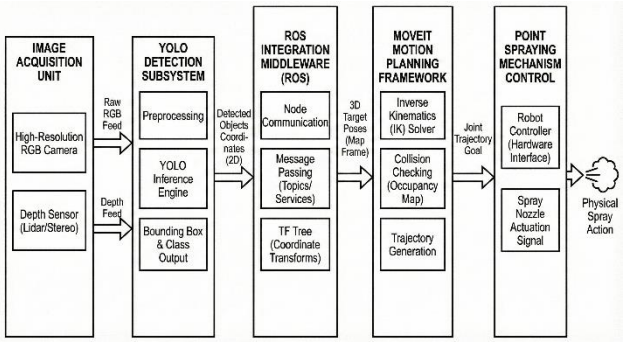

Figure 8. Overall system architecture.

Results

The performance of the developed framework was evaluated at two levels. First, the performance of the trained AI model was assessed, as it directly affects the

performance of the overall system. Subsequently, the performance of the complete system was evaluated after confirming the successful operation of the trained AI model.

After training the AI model using 450 images from the 526-image dataset, the remaining 76 (50 diseased, 26 healthy plant) images were used to test the model's disease detection performance on leaf images. In 63 out of 76 test cases, the AI model produced correct outputs, corresponding to an approximate success rate of 83%. This value represents the upper performance limit of the overall system.

Before evaluating the full system performance involving both the mobile base and the robotic arm, an intermediate test was conducted. In this step, the ability of the robotic arm to scan a plant, identify the diseased area, and reach the target end-effector position without the mobile base was evaluated. Based on these tests, improvements were made to the algorithms responsible for generating the target end-effector position. A link to a sample video demonstrating this application is provided in the Appendix.

To evaluate the performance of the complete framework, five leaf models with diseased areas were created in the simulation environment and placed within plant models. The simulation was conducted 20 times for each of three different field setups. In each setup, the plant models were positioned at different locations and orientations within the field. The configuration of each setup remained unchanged throughout the 20 test iterations.

In the simulation environment, the robot manipulator was required to scan plants and identify diseased regions, which is a more challenging task than detecting diseases in static, pre-captured images. The disease detection success rate in the simulation environment was 75%, which aligns with expectations. Among the cases in which disease detection was successful, the disease localization success rate was 80%, and the successful execution rate of agricultural spraying was 75%. Consequently, the cumulative success rate of the overall process was calculated as 56.25%. Links to a sample video demonstrating the overall framework performance and to the Git repository of this study are provided in the Appendix. The success rates of individual process steps across the three simulation setups are given in Table 3.

Table 3: Success Rates of Process Steps in Simulation

Steps	Setup 1	Setup 2	Setup 3
Disease Detection (%)	78	74	73
Disease Localization (%)	82	79	79
Spraying (%)	76	74	75
Cumulative (%)	58	55	56

Table 4 presents a comparative overview of selected studies on robotic systems for plant disease detection and spraying. As shown, many existing studies primarily focus on disease detection or plant recognition, whereas spraying—when included—is often performed on the entire plant rather than at the level of the diseased area. Spot spraying is addressed in only a limited number of studies. A direct comparison of reported success rates is not straightforward, as these values are based on different evaluation metrics and experimental conditions. While some studies consider only disease detection [22]–[24], others incorporate spraying either on the diseased area [16] or on the entire plant [13], [14]. The study most closely related to our work [16], which also includes spot spraying, focuses on powdery mildew and evaluates performance based on the percentage of sprayed area relative to the total diseased area. In contrast, our study considers multiple disease types and reports the success rate of the trained model in detecting diseased regions. Since spot spraying involves several sequential steps, errors may accumulate across the pipeline, leading to a reduced overall system performance, as reflected in the results summarized in Table 3.

Table 4: Comparison of Studies with Mobile Base in Literature

Studies	Disease Detection	Spraying	Target Disease/Plant	Success Rate
[13]	Plant Recognition	Whole Plant	Tabacco	~92%
[14]	Yes	Whole Plant	Multiple Diseases	95% -97%
[15]	Yes	Yes	Spodoptera Littorals	N/A
[16]	Yes	Spot	Powdery Mildew	85%-100%
[22]	Yes	No	Color Change on Leaves	N/A
[23]	Yes	No	Rust, Scab	>95%
[24]	Yes	No	Distinction of Bacterial, Viral or Late Blight	~92% - ~98%
This Work	**Yes**	**Spot**	**Chlorosis, Powdery Mildew, Septoria**	**83.0%**

Conclusion

Autonomous agricultural spraying is a multi-stage process in which the success of each step depends directly on the performance of the preceding one; therefore, a relatively low cumulative success rate is expected. Despite this inherent complexity, this study demonstrates the feasibility of an integrated framework that combines AI-based plant disease detection, 3D localization, and autonomous spot spraying using a mobile robot manipulator, representing a novel and comprehensive approach within the existing literature.

Due to the lack of a ready-to-use dataset containing images of whole plants with realistic disease symptoms, a custom dataset was created through web scraping and manual image acquisition. While the limited dataset size affected the overall performance of the framework, the results validate the effectiveness of the proposed methodology. Data augmentation was deliberately not applied, as the available data was insufficient to generate realistic symptom variations without introducing bias.

One of the key technical contributions of this study is the geometric method proposed for calculating the surface normal of diseased areas using RGB-D data, which enables precise end-effector positioning for spot spraying at an optimal orientation. Unlike many existing approaches that rely on whole-plant spraying, the proposed system performs localized, surface-aware spraying, significantly reducing unnecessary pesticide usage. This spot-spraying strategy represents a meaningful advancement toward environmentally sustainable precision agriculture.

Initially, a 5-axis robotic arm was considered, since rotation around the nozzle's z-axis does not affect spraying performance. However, inverse kinematics solutions were not consistently achievable. The adoption of a 6-axis robotic arm successfully resolved this limitation, highlighting the importance of kinematic redundancy for autonomous agricultural manipulation tasks. Although collision avoidance was not implemented in the current framework, future work will integrate collision avoidance strategies to prevent damage to surrounding plant structures. Such functionality may require redundant manipulators and advanced motion planning techniques.

The joint configuration of the robotic arm is computed based on the mobile base position at the moment of plant scanning and disease detection. In some scenarios, inverse kinematics fails to produce a feasible or collision-free solution. As a future extension, task prioritization and coordinated base–manipulator motion will be investigated to dynamically reposition the mobile platform and improve reachability.

For simplicity, a two-wheeled mobile base was used in this study. However, real-world deployment would require a four-wheeled or tracked platform for improved stability on uneven terrain. Furthermore, real agricultural environments introduce challenges such as variable lighting, dust, and terrain irregularities. Addressing these challenges will require sensor fusion techniques and robust localization methods.

Overall, this study contributes a unique, end-to-end robotic framework that integrates disease detection, 3D localization, and autonomous spot spraying within a unified ROS-based architecture. The proposed system advances the state of the art by moving beyond detection-only approaches and demonstrating the practical application of mobile manipulators for precision agricultural spraying. The results highlight the potential of autonomous robotic systems to reduce labor costs, minimize chemical usage, and improve sustainability in modern agriculture.

References

[1] I. Ghosh, "Mapped: The dramatic global rise of urbanization (1950-2020)," *World Economic Forum*, Feb. 9, 2019. [Online]. Available: https://www.weforum.org/agenda/2019/09/mapped-the-dramatic-global-rise-of-urbanization-1950-2020/

[2] A. A. Akram, S. Chowdhury, and A. M. Mobarak, "Effects of emigration on rural labor markets," *National Bureau of Economic Research*, Working Paper 23929, Oct. 2017. [Online]. Available: http://www.nber.org/papers/w23929

[3] Bilgi Teknolojileri ve İletişim Kurumu (BTK), "Araştırma raporları," 2019. [Online]. Available: https://btk.gov.tr/arastirma-raporlari. Accessed: Dec. 16, 2025.

[4] European Commission, *Farm to Fork Strategy: For a Fair, Healthy and Environmentally-Friendly Food System*, COM(2020) 381 final, Brussels, Belgium, May 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020DC0381

[5] A. Al Kuwaiti, K. Nazer, A. Al-Reedy, S. Al-Shehri, A. Al-Muhanna, A. V. Subbarayalu, D. Al Muhanna, and F. A. Al-Muhanna, "A review of the role of artificial intelligence in healthcare," *Journal of Personalized Medicine*, vol. 13, no. 6, Art. no. 951, Jun. 2023. [Online]. Available: https://doi.org/10.3390/jpm13060951

[6] C. Maraveas, "Incorporating artificial intelligence technology in smart greenhouses: Current state of the art," *Applied Sciences*, vol. 13, no. 1, p. 14, 2022. [Online]. Available: https://doi.org/10.3390/app13010014

[7] J. Pedersen and K. M. Lind, Precision Agriculture and the Future of Farming in Europe, European Parliament, 2017. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/STUD/2016/581892/EPRS_STU(2016)581892_EN.pdf

[8] O. Karaçay and S. Kılıç, "The effects of agricultural tire technologies on soil compaction, traction performance and agricultural productivity," *Scientific Journal of Mehmet Akif Ersoy University*, vol. 7, no. 2, pp. 64–80, 2024. [Online]. Available: https://doi.org/10.70030/sjmakeu.1563596

[9] A. J. Hamza and W. L. Anderson, "A review on the effect of soil compaction and its management for sustainable crop production," J. Biosystems Eng., vol. 46, pp. 417–439, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s42853-021-00117-7

[10] E. Anastasiou, S. Fountas, M. Voulgaraki, V. Psiroukis, M. Koutsiaras, O. Kriezi, et al., "Precision farming technologies for crop protection: A meta-analysis," Smart Agricultural Technology, vol. 5, Art. no. 100323, 2023. [Online]. Available: https://doi.org/10.1016/j.atech.2023.100323

[11] R. Sapkota, J. Stenger, M. Ostlie, *et al.*, "Towards reducing chemical usage for weed control in

agriculture using UAS imagery analysis and computer vision techniques," *Scientific Reports*, vol. 13, Art. no. 6548, 2023. [Online]. Available: https://doi.org/10.1038/s41598-023-33042-0

[12] M. İnan and A. Karcı, "Tarımda ağaç ilaçlamanın drone'larla yapılmasında yeni bir yöntemin geliştirilmesi ve uygulanması," *Computer Science*, vol. 6, no. 2, pp. 72–89, 2021. [Online]. Available: https://izlik.org/JA96YF74TF

[13] F. E. Nasir, M. Tufail, M. Haris, J. Iqbal, S. Khan, and M. T. Khan, "Precision agricultural robotic sprayer with real-time tobacco recognition and spraying system based on deep learning," *PLOS ONE*, vol. 18, no. 3, Art. no. e0283801, Mar. 2023. [Online]. Available: https://doi.org/10.1371/journal.pone.0283801

[14] S. Forhad, K. Z. Tayef, M. Hasan, A. N. M. S. Hasan, M. Z. Islam, and M. R. K. Shuvo, "An autonomous agricultural robot for plant disease detection," in *The Fourth Industrial Revolution and Beyond: Select Proc. IC4IR+*, Lecture Notes in Electrical Engineering, vol. 980, Singapore: Springer Nature Singapore, 2023, pp. 695–708. [Online]. Available: https://doi.org/10.1007/978-981-19-8032-9_50

[15] M. Al-Mamury, N. Manivannan, H. Al-Raweshidy, and W. Balachandran, "Mobile robot based electrostatic spray system for controlling pests on cotton plants in Iraq," *Journal of Physics: Conference Series*, vol. 646, no. 1, p. 012008, 2015. [Online]. Available: https://doi.org/10.1088/1742-6596/646/1/012008

[16] R. Oberti, M. Marchi, P. Tirelli, A. Calcante, M. Iriti, E. Tona, M. Hočevar, J. Baur, J. Pfaff, C. Schütz, and H. Ulbrich, "Selective spraying of grapevines for disease control using a modular agricultural robot," *Biosyst. Eng.*, vol. 146, pp. 203–215, 2016. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2015.12.004

[17] D. T. Fasiolo, L. Scalera, E. Maset, and A. Gasparetto, "Towards autonomous mapping in agriculture: A review of supportive technologies for ground robotics," *Robot. Auton. Syst.*, vol. 169, Art. no. 104514, 2023. [Online]. Available: https://doi.org/10.1016/j.robot.2023.104514

[18] H. Özgen and M. Turan, "Sulama/ilaçlama robotu için nesne tanıma çalışmaları," *Avrupa Bilim ve Teknoloji Dergisi (EJOSAT Ek Özel Sayı HORA)*, pp. 25–33, 2021. [Online]. Available: https://doi.org/10.31590/ejosat.1115830

[19] S. Ramesh and R. Hebbar, "Plant disease detection using machine learning," in *Proc. 2018 Int. Conf. on Design Innovations for 3Cs Compute, Communicate, Control (ICDI3C)*, Bangalore, India, Apr. 2018, pp. 41–45. [Online]. Available: https://doi.org/10.1109/ICDI3C.2018.00017

[20] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, Art. no. 1419, Nov. 2016. [Online]. Available: https://doi.org/10.3389/fpls.2016.01419

[21] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, Nov. 2019. [Online]. Available: https://doi.org/10.3390/plants8110468

[22] P. S. Kesava and K. P. Peeyush, "Autonomous robot to detect diseased leaves in plants using convolutional neural networks," in *Proc. 3rd Int. Conf. Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, Apr. 2019, pp. 806–809. [Online]. Available: https://doi.org/10.1109/ICOEI.2019.8862737

[23] P. Karpyshev, V. Ilin, I. Kalinov, A. Petrovsky, and D. Tsetserukou, "Autonomous mobile robot for apple plant disease detection based on CNN and multi-spectral vision system," in *Proc. 2021 IEEE/SICE Int. Symp. on System Integration (SII)*, Iwaki, Fukushima, Japan, Jan. 2021, pp. 157–162. [Online]. Available: https://doi.org/10.1109/IEEECONF49454.2021.9382649

[24] A. Xenakis, G. Papastergiou, V. C. Gerogiannis, and G. Stamoulis, "Applying a convolutional neural network in an IoT robotic system for plant disease diagnosis," in *Proc. 11th Int. Conf. Information, Intelligence, Systems and Applications (IISA)*, Piraeus, Greece, Jul. 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1109/IISA50023.2020.9284356

[25] I. Hrabar, G. Vasiljević, and Z. Kovačić, "Estimation of the energy consumption of an all-terrain mobile manipulator for operations in steep vineyards," *Electronics*, vol. 11, no. 2, p. 217, 2022. [Online]. Available: https://doi.org/10.3390/electronics11020217

[26] H. Hejazipoor, J. Massah, M. Soryani, K. A. Vakilian, and G. Chegini, "An intelligent spraying robot based on plant bulk volume," *Computers and Electronics in Agriculture*, vol. 180, Art. no. 105859, 2021. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105859

[27] K. V. N. Rajesh and D. L. Bhaskari, "Optimum dataset size for ayurvedic plant leaf recognition using convolution neural networks," in *Data Management, Analytics and Innovation: Proc. ICDMAI 2021*, vol. 1, Singapore: Springer, 2021, pp. 249–261. [Online].

Available: https://doi.org/10.1007/978-981-16-2934-1_16

[28] V. Ponnusamy, A. Coumaran, A. S. Shunmugam, K. Rajaram, and S. Senthilvelavan, "Smart glass: Real-time leaf disease detection using YOLO transfer learning," in *Proc. 2020 Int. Conf. Commun. Signal Process. (ICCSP)*, Jul. 2020, pp. 1150–1154. [Online]. Available: https://doi.org/10.1109/ICCSP48568.2020.9182146

[29] Pjreddie, "YOLO: Real-Time Object Detection," 2018. [Online]. Available: https://pjreddie.com/darknet/yolo/

[30] Stanford University, "Convolutional neural networks (CNNs / ConvNets)," 2018. [Online]. Available: https://cs231n.github.io/convolutional-networks/

Appendix

Link to the video of agricultural spraying with just the robotic arm: https://www.youtube.com/watch?v=Z2POisrO0QU

Link to a sample video of the simulation of the mobile manipulator navigating in the field, scanning the plants and performing agricultural spraying: https://www.youtube.com/watch?v=wDgrFnwk_xY

Git repository of the project: https://github.com/pyimagedata/mobile_manipulator