International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2025, 9 (1), 151-161
DOI: 10.36287/ijmsit.9.1.21
Received: June 18, 2025; Accepted: July 27, 2025

Detection and Disinfestation of Diseased Plants with YOLO Based ANFIS Controlled Unmanned Ground Vehicle

Serhat YILMAZ[1*], Dilara POLAT[2], Esma Beyza AKBOYNUZ[3], Emir Alp GEDİKLİ[4], Zeynep YILMAZ[5]

[1] Department of Electronics and Communication, Engineering Faculty, Kocaeli University, Kocaeli, Türkiye (serhaty@kocaeli.edu.tr) (ORCID: 0000-0001-9765-7225)*
[2]Department of Electronics and Communication, Engineering Faculty, Kocaeli University, Kocaeli, Türkiye (dlr012@outlook.com) (ORCID: 0009-0006-9978-0896)
[3]Department of Electronics and Communication, Engineering Faculty, Kocaeli University, Kocaeli, Türkiye (esmabeyzaakboynuz@gmail.com) (ORCID: 0009-0002-1740-4290)
[4]Department of Electronics and Communication, Engineering Faculty, Kocaeli University, Kocaeli, Türkiye (alp_gedikli@hotmail.com) (ORCID: 0009-0001-3882-2873)
[5]Faculty of Engineering and Natural Sciences , Materials Science and Nano Engineering, Sabancı University, Tuzla/Istanbul, Türkiye (zeynep.yilmaz@sabanciuniv.edu) (ORCID: 0009-0006-0546-0411)

Abstract – Plant diseases remain a major challenge in modern agriculture, causing considerable reductions in both yield and crop quality. This study focuses on the development of an intelligent unmanned ground vehicle (UGV) capable of detecting plant diseases in real time and autonomously responding through targeted spraying. A camera mounted on the UGV captures continuous images of crop rows, and disease detection is carried out using the YOLO (You Only Look Once) algorithm—chosen for its speed and accuracy in real-time object recognition. To evaluate model performance, YOLOv7, v8, and v9 were trained using datasets focused on potato leaf diseases, including early and late blight. The YOLOv8 model was selected for deployment on a Raspberry Pi 4B based on its superior detection accuracy. Additionally, a servo motor-enhanced vision system was implemented to broaden the camera's coverage.
The UGV's autonomous driving is enabled by a combination of five ultrasonic sensors and an ANFIS (Adaptive Neuro-Fuzzy Inference System)-based decision-making module, which governs navigation and motion planning. As the vehicle traverses the field, the onboard system identifies infected plants and activates a localized spraying mechanism to treat only the affected areas. This integrated approach significantly reduces pesticide use, minimizes environmental harm, and lowers the dependency on manual labor. The results demonstrate a promising application of artificial intelligence and embedded systems for sustainable and efficient disease management in precision agriculture.

Keywords – *Diseased Plant Detection, YOLO, Spraying, ultrasonic sensor, Raspberry Pi, motion planning, ANFIS*

Citation: Yılmaz et. Al. (2025). Detection and Disinfestation of Diseased Plants with YOLO Based ANFIS Controlled Unmanned Ground Vehicle. International Journal of Multidisciplinary Studies and Innovative Technologies, 9(1): 151-161.

I. Introduction

Agriculture, one of humanity's most vital sectors, faces significant challenges due to plant diseases and pests, which threaten the sustainability of food production. These diseases, caused by various fungi, pathogens, viruses, or bacteria, severely affect both the yield and quality of crops. Traditional methods of detecting plant diseases are often time-consuming and inefficient. However, advancements in artificial intelligence (AI) and image processing technologies have introduced innovative solutions for early and accurate disease detection [1]. The excessive use of pesticides and agrochemicals has further underscored the need for precise disease identification and targeted spraying. Recent research has focused on developing autonomous systems capable of real-time plant disease detection and treatment. Many studies have employed deep learning algorithms, particularly the YOLO family of object detection models, due to their speed and accuracy in recognizing diseased plant areas [2]–[7].

For instance, Shill [2] utilized YOLOv3 and YOLOv4 algorithms to classify 17 different disease types across 13 plant species, while Soeb et al. [3] applied the YOLOv7 technique for detecting five types of tea leaf diseases using a dataset of 4000 images. Mahesh [4] implemented YOLOv5 for early detection of bacterial spots on bell peppers, aiming to assist farmers with timely interventions. Similarly, Reddy [5] focused on detecting diseases on mulberry leaves critical to the sericulture industry using YOLO combined with Convolutional Neural Networks (CNNs).

Other notable works include Singh and Misra's [6] research on image segmentation for plant disease detection using MATLAB, and Dung's [7] development of a real-time disease detection system by implementing a Single Shot Detector (SSD) model on a Raspberry Pi 3 platform. Though Singh and Misra's algorithm achieved high accuracy, its slow processing limited its use in real-time applications. In contrast, Dung's model delivered faster and reliable detection suitable for real-world scenarios.

*Corresponding Author:Serhat YILMAZ, e-mail: serhaty@kocaeli.edu.tr, Tel: 05319625820

Furthermore, researchers such as Sathiesh Kumar [8] developed a rover, incorporating a Raspberry Pi 2 board and ATmega2560 microcontroller to collect environmental data (temperature, humidity, and weather conditions) and assist in autonomous navigation using ultrasonic sensors. Sharanesh et al. [9] employed VGGNet16 and InceptionResNetV2 architectures to detect diseases in crops like strawberries and grapes, achieving a success rate of 95.23%. Ahmed et al. designed Agrobot, a smart agricultural monitoring system that operates in real time. The system utilizes a Raspberry Pi for low-cost hardware implementation and employs the YOLO algorithm for detecting plant health conditions and other agricultural features. Field experiments demonstrated the system's potential for efficient and automated crop monitoring [10]. Another study presents the development of a pesticide spraying robot integrated with IoT technology for agricultural applications. The robot aims to minimize manual labor and optimize the spraying process through remote control and monitoring. Its design focuses on improving safety, precision, and coverage in pesticide application [11]. Vani et al. proposed a machine learning method for detecting, tracking, and forecasting plant diseases to help farmers manage crop health effectively. The approach demonstrates promising accuracy in early disease identification and progression prediction [12]. Building on recent advancements in smart agriculture, this study proposes an autonomous Unmanned Ground Vehicle (UGV) designed to detect diseases in potato plants and perform targeted spraying in real time. The UGV autonomously navigates between crop rows using a set of ultrasonic sensors, and identifies diseased plants through deep learning-based image processing running on a Raspberry Pi platform. Upon detection, a precision spraying mechanism is activated to treat only the affected plants.

II. SYSTEM OVERVIEW

This integrated system—combining motion planning, autonomous navigation, intelligent disease identification, and localized disinfestation—aims to reduce pesticide usage, lower labor demands, and improve both the quality and yield of potato crops. The overall design promotes and contributes to efficient, scalable, and environmentally friendly agricultural practices (Fig.1).

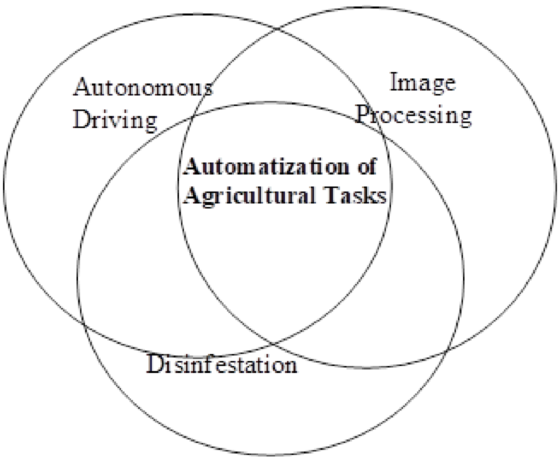

Fig.1. Conceptual Purposes of the Designed UGV System

The UGV navigates autonomously using five ultrasonic sensors and an ANFIS-based control system on an Arduino UNO. A Raspberry Pi processes camera images in real time to detect plant diseases using a YOLO model. A spraying mechanism targets infected areas, and four DC motors provide movement via PWM control (Fig.2). The system reduces pesticide use which improves crop quality, as will be illustrated in experimental tests.

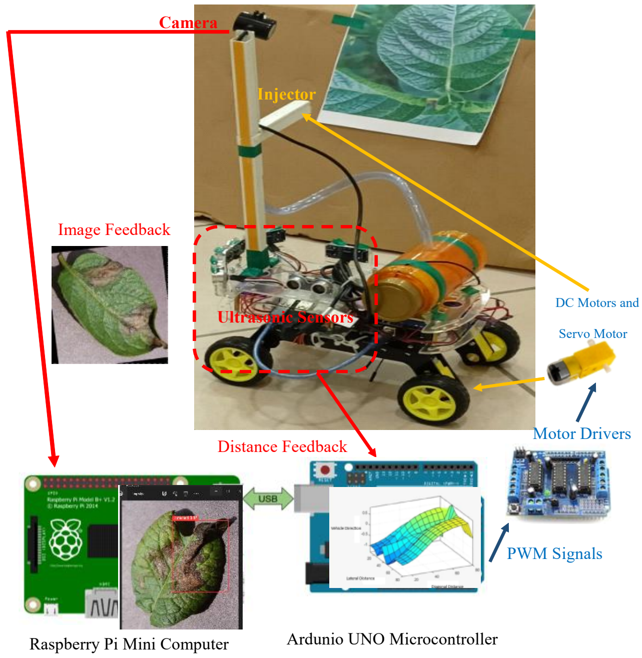

Fig.2. Hardware and software features which provide the vehicle operation

III. DISEASE DETECTION WITH YOLOV8 ALGORITHM

Image processing involves extracting meaningful information from digital images and is widely applied in fields like computer vision, pattern recognition, and object detection. It also plays a key role in industries such as healthcare, agriculture, automotive, security, and media. YOLO is a deep learning-based object detection algorithm that processes an image in a single pass, delivering faster and more efficient results compared to traditional methods. For instance Hamza et al. compared modified YOLO v5 and Faster R-CNN for recycling waste classification. Both models were evaluated on accuracy and speed using waste image datasets. Results show that the modified YOLO v5 offers faster inference with competitive accuracy, while Faster R-CNN achieves slightly higher accuracy but at slower speeds. The findings suggest that modified YOLO v5 is better suited for real-time waste classification applications [13]. In an other study a novel approach for multimodal medical image segmentation by combining the YOLOv8 object detection framework with SAM and HQ-SAM segmentation models. The method improves segmentation accuracy across diverse medical imaging modalities [14].

Its ability to detect all objects simultaneously makes it highly suitable for real-time applications like autonomous vehicles, surveillance systems, and security solutions.

A. YOLOV8 Training Procedure

In the study, the Yolov8 algorithm [15] was used for real-time plant detection. The model was trained with the Plant Village and New Plant Disease datasets [5,16]. The goal was to detect early blight and late blight diseases in plants using the model. The number of data points used from the relevant datasets is provided in the tables below.

Table 1. Plant Village Data Set [5]

Disease Type	Training Data	Validation Data
Early Blight	900	100
Late Blight	900	100

Table 2. New Plant Disase Data Set [16]

Disease Type	Training Data	Validation Data
Early Blight	3872	970
Late Blight	3878	970

In this study, both the PlantVillage and the New Plant Disease datasets were utilized to enhance model performance and improve generalization capability. These datasets were combined during the training process to increase the diversity and volume of samples, particularly for early and late blight disease classes. By integrating images from both datasets—collected under varied lighting conditions, resolutions, and backgrounds—the model was exposed to a broader range of visual features, allowing it to generalize better to real-world field conditions. All images in the datasets had already been standardized in the databases in size and format, and were annotated in YOLO format to ensure consistency across the combined dataset. In this study, data sets are directly combined, used in training and validation of the model.

A.1. Image Processing and YOLOv8 Training Stages

Standard YOLOv8 training includes preprocessing steps like image resizing and normalization, along with automatic data augmentation techniques such as horizontal flipping, brightness and HSV color adjustments, mosaic augmentation, and random cropping or translation. These augmentations help manage data variability caused by lighting differences, viewing angles, and occlusions, enabling the model to generalize well to diverse real-world conditions. Additionally, the training datasets contained images captured under varied environmental settings, further improving the model's robustness against such variabilities. YOLO models typically expect square input images like 416×416 or 640×640. The training framework usually handles these transformations automatically. Ultralystics library is used under YOLO for object detection, classification, segmentation and pose estimation of an image. In this study Yolov7, Yolov8 and Yolov9 models are tested. Training and validation data sets are obtained by data collection, labelling, object detection stages. The stages are explained as follows;

Dataset Collection: Images were sourced from Kaggle using keywords like "Fungi", "Early Blight", and "Late Blight", mainly using PlantVillage and New Plant Disease datasets.
Labeling: Images are labeled as "defective" with labelImg under Roboflow application and are saved in YOLO format.
Augmentation & Split: YOLOv8 is installed, trained and run using the Anaconda Prompt command line. The data sets are augmented, split, uploaded to Google Drive, and used in Google Colab.

Object Detection: Image is split into an N×N grid. The model predicts object locations and confidence scores (Fig.3).

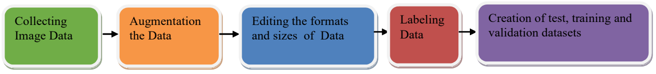

Fig.3. Processing of Potato Leaves Image Data

The following figures show the detection and labeling stages of a leaf image, respectively in this study (Fig.4 and Fig.5). YOLOv8 includes several structural improvements compared to YOLOv5 and YOLOv7. The integration of the C2f module enhances the model's ability to extract meaningful features, which is particularly helpful when working with small or subtle patterns. In addition, the move to an anchor-free detection head allows for more adaptive object recognition across varying scales. These design changes have proven beneficial in our case, where accurately identifying small disease signs is essential [17]

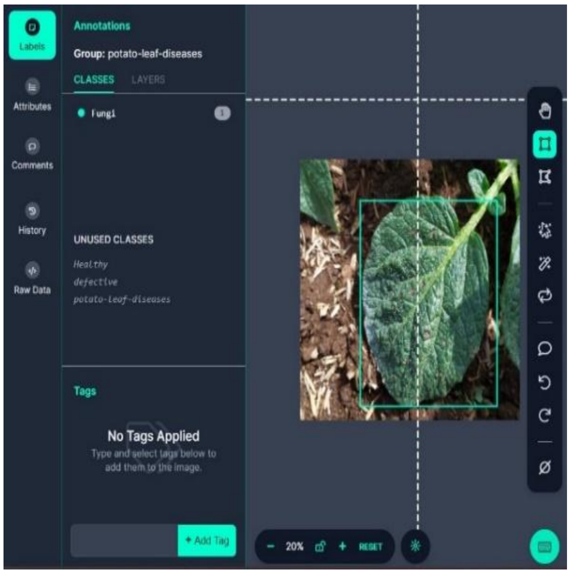

Fig.4. Detection and Framing of the Leaf Image

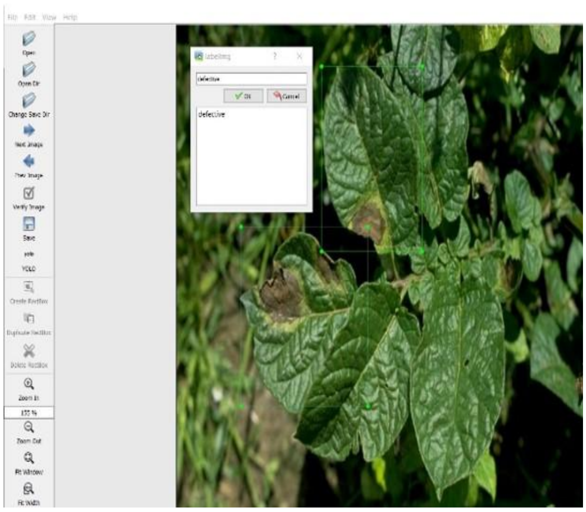

Fig.5. Labeling of a Leaf Image

The YOLO model's training graphs based on the data in Tables 1 and 2 are shown in Fig. 6

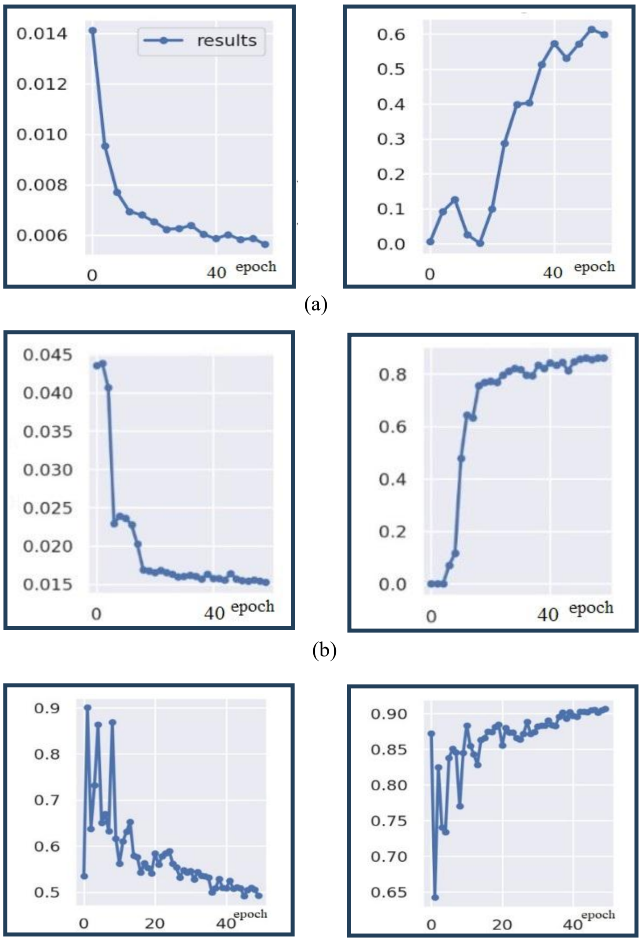

Fig.6. Comparison Results of Training Models for Validation Loss (val_loss) and Average Precision (mAp@0,05) Graphics
(a) Yolov7, (b) Yolov8, (c) Yolov9

Among the trained models, YOLOv9 achieves the highest average accuracy (mAP@0.5 > 90%), while YOLOv7 exhibits the lowest validation loss. YOLOv8, on the other hand, demonstrates a more balanced and stable performance overall. As a result of the training, for the proposed YOLOv8 method it was observed that the disease detection accuracy of the model on the data was 90%.

A.2. Execution of Real Time Disase Detection Stage

The execution of the real-time disease detection stage involves a structured pipeline designed to perform plant disease recognition directly from live video input. Initially, all required libraries are imported, and appropriate paths for model files and resources are configured. Subsequently, the YOLO model is loaded into the system (Fig.7). A webcam interface is activated to capture frames in real time, which are then processed by the YOLO model to estimate the presence of disease indicators. The model applies automatic inference techniques and returns prediction results in the form of bounding box coordinates, associated class labels, and confidence scores. These outputs are used to annotate the live video feed by drawing labeled bounding boxes around the detected regions (Fig.8). This approach facilitates robust, real-time identification of plant diseases, enabling immediate visual feedback and potential integration into autonomous agricultural systems.

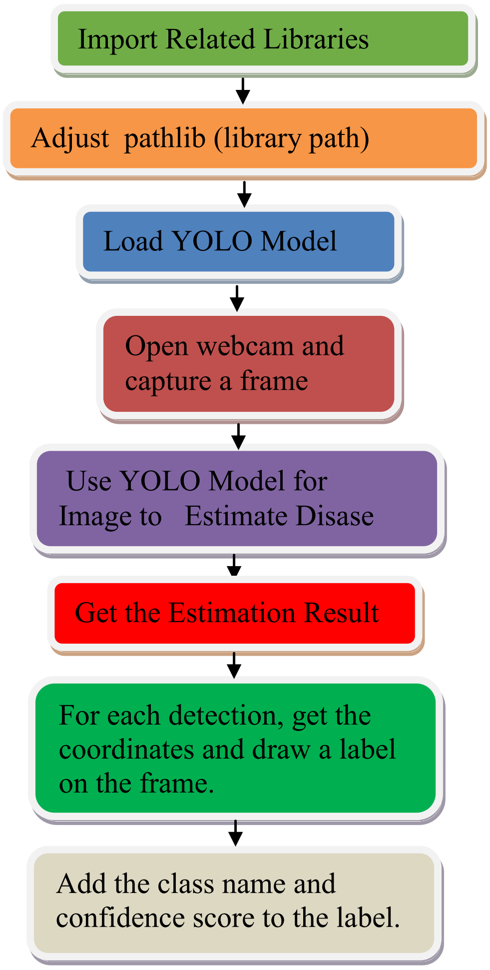

Fig. 7. YOLOv8-based Real-Time Plant Disease Detection Algorithm Flowchart

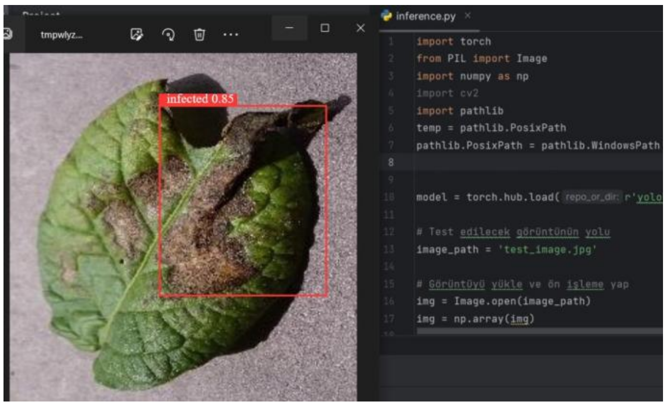

Fig.8. Plant Disease Detection and Labeling Stages

B. UGV Motion Planning and Autonomous Driving System

The UGV, equipped with fully autonomous driving capabilities, is designed to navigate from a starting point to a target destination while avoiding both static and dynamic obstacles and performing designated tasks along the route. To accomplish this, the vehicle must accurately process positional and environmental data obtained from sensors [8].

Accordingly, its path planning, navigation, decision-making, and control systems have been developed to ensure reliable motion planning and mission completion without collisions. The autonomous motion planning of the UGV is illustrated in Fig.9.

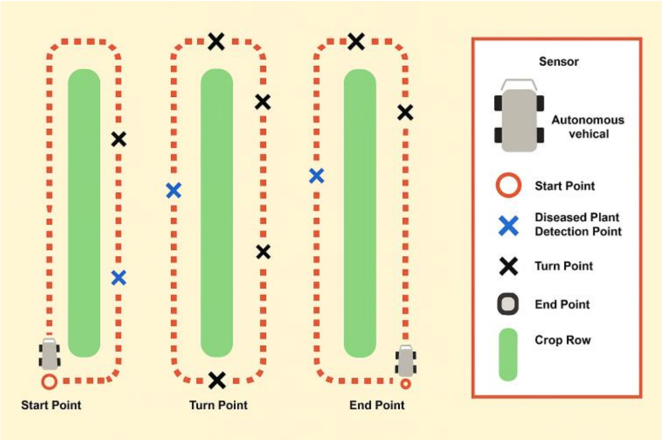

Fig.9. UGV Motion Planning

The proposed navigation strategy enables the UGV to autonomously traverse crop rows while maintaining optimal alignment and performing targeted interventions. The vehicle follows each row starting from a predefined point, maintaining a lateral distance of M1 = 10 cm from the crops and using a diagonal reference of M2 = 20 cm to ensure parallel alignment (Fig.10). On odd-numbered rows, right-side sensors are active, while on even-numbered rows, left-side sensors are used. At the end of each row detected by cross-mounted sensors, the UGV executes a turn: right turns after odd rows and left turns after even rows, enabling continuous coverage of the field. When a diseased plant is detected, the vehicle halts and applies treatment before resuming navigation. To support robust operation in realistic environments, including uneven terrain and curved rows, a comprehensive path planning algorithm was implemented. This algorithm ensures the UGV adheres closely to plant rows and successfully transitions between them, optimizing both disease intervention and field coverage from start to finish.

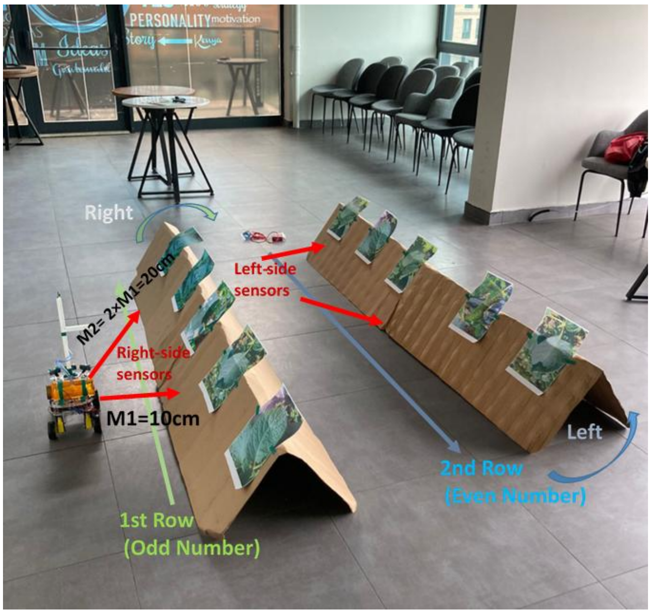

Fig.10. Navigation of the UGV in Laboratory Environment

These scenarios are presented in Figures 11 and 12, respectively.

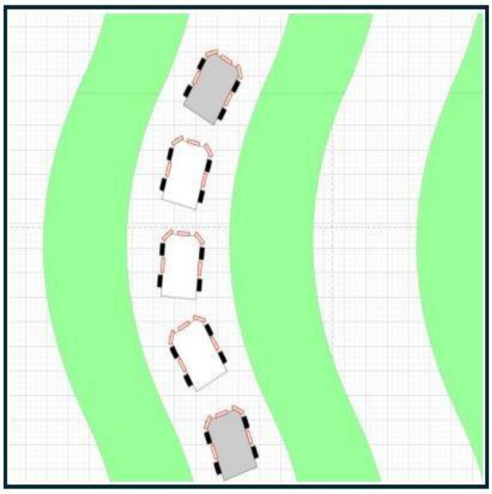

Fig 11. UGV motion scenario on an uneven road

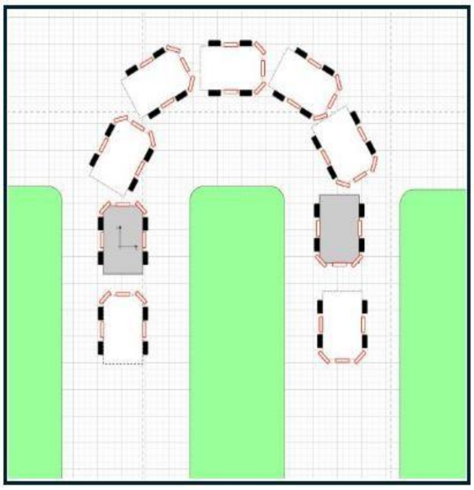

Fig.12. UGV Turning Scenario at the End of the Crop Row.

Fig.12 considers the scenario where the IGA travels in a non-flat row of crops in the field. In this scenario, the aim is to prevent the UGV from leaving the row of crops and to ensure that it moves at an appropriate distance from the row of crops for spraying. In the scenarios design of the navigation system aims to ensure that the UGV maintains its position and can continue its mission smoothly. The goal is for the guidance unit to produce accurate reference values and to derive information by accurately receiving and processing data from the navigation system.

C. Navigation Module

The aim of the UGV is to perceive the vehicle's environment, determine its position, create a route plan and perform its task according to this route through the navigation system. In this study, an ultrasonic sensor and camera module were selected to implement the navigation system.

C.1. Sensor Configuration and Navigation Strategy

The UGV's navigation relies on a set of five ultrasonic sensors (U1–U5) strategically positioned to ensure precise alignment with crop rows and effective maneuvering, as illustrated in Fig.13. The angle between the U1 and U2 sensors is set at 60°. U1 and U5, located on the right and left sides respectively, measure the lateral distance between the vehicle and the crop rows. U2 and U4, positioned diagonally, estimate the vehicle's turning angle (ranging from −45° to +45°) and help detect the end of the row. Depending on the row number, only the sensors on the active side are used—right for odd-numbered rows, and left for even-numbered rows—to maintain optimal alignment. U3, placed at the front, provides obstacle detection and assists in decision-making during turning maneuvers (as detailed in Table 3). This sensor setup enables reliable positioning and smooth transitions between crop rows.

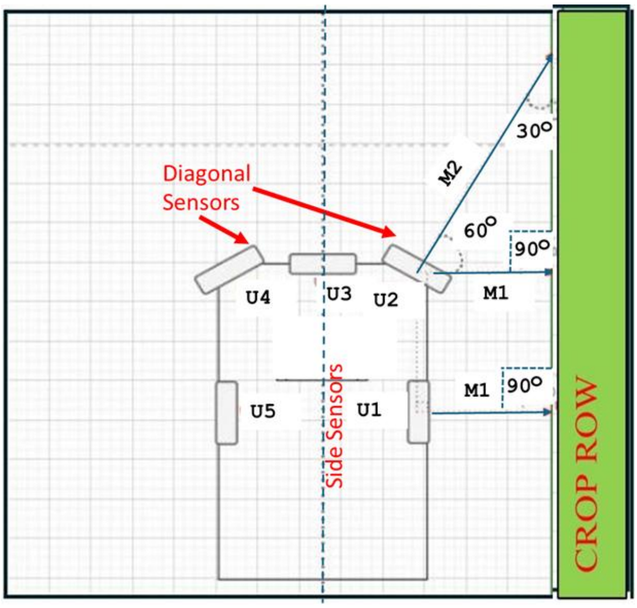

Fig.13. Sensor Configuration

Table 3. Sensors Operation

Sensor	Position	Function
U1	Right Side	Measures distance to the crop row (M1)
U2	Right Diagonal	Determines turning angle and row end (M2)
U3	Front	Detects front obstacles
U4	Left Diagonal	Determines turning angle and row end (M2)
U5	Left Side	Measures distance to the crop row (M1)

With the aid of the camera module, the UGV performs its mission of detecting diseased plants. The ultrasonic sensors, on the other hand, serve the function of a lidar system. Due to the 30-60-90 triangle formed in this configuration, the M2 distance is calculated to be twice the M1 distance [6]. In every scenario where this configuration occurs, the UGV moves straight within the crop row.

C.2. Algorithm for Autonomous Navigation and Spraying

This algorithm defines the control process for an Unmanned Ground Vehicle (UGV) designed to autonomously navigate through agricultural crop rows, detect diseased plants, and perform targeted spraying. The operation begins with system initialization, followed by the activation of relevant sensors based on the current row number. The UGV then moves forward through the row while continuously monitoring for plant health. When a diseased plant is detected, the system triggers the spraying mechanism. At the end of each row, a check is performed to determine whether the row has been fully traversed. Based on the current row number, the UGV makes a directional decision to turn into the next row. This cycle continues until all rows are completed, upon which a final check confirms the completion of the entire field. The algorithm diagram is provided in Figure 14.

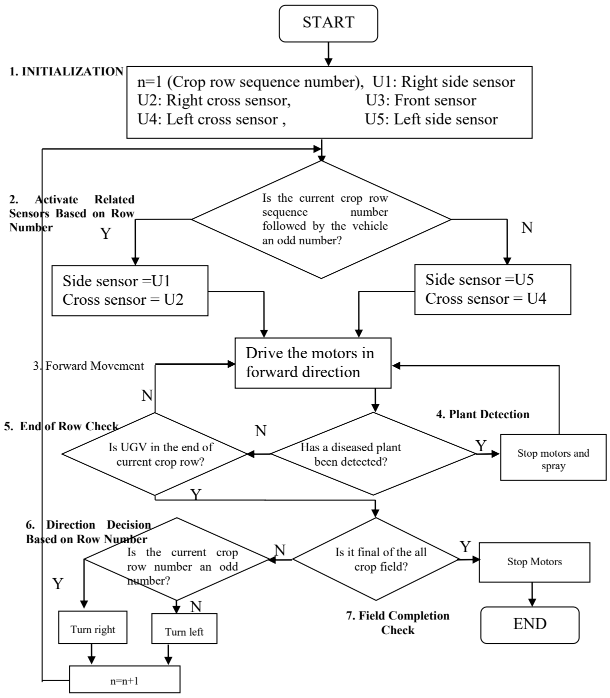

Fig.14 UGV Path Planning Algorithm

D. Inference Module (ANFIS)

Adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS) is a type of artificial neural network based on the Takagi-Sugeno fuzzy inference system. The technique was developed in the early 1990s [18,19]. Since it combines both neural networks and fuzzy logic principles, it has the potential to capture the benefits of both in a single framework.

The outputs of the path planning algorithm and navigation module were used to ensure that the UGV maintains its position and successfully completes its mission. The data processed in the guidance module played a role in generating reference values for the UGV's motion control [20]. The possible positions of the UGV relative to the crop row are given in Table.4. For example, Scenario M2 > 2*M1 describes that the UGV moving away from the crop to the left. M2 = 2*M1 is the ideal situation. Because based on the sensor placement, when the vehicle moves parallel to the row, the distance M2 should be twice the distance M1. For the Scenario of 2*M1> M2 > M1, UGV approaches the crop (to right) too close from front. For the Scenario M2 < M1, UGV approaches the crop too close from back.

Table 4. Lateral and Diagonal Sensor Value Range Based on the UGV's Rotation Angle

Rotation Angle (º)	Scenario According to Sensor Values
-45- 0	M2 > 2*M1
0	M2 = 2*M1
0-15	2*M1> M2 > M1
15-45	M2 < M1

When the UGV encounters positional deviations, it autonomously adjusts its orientation within the row using an ANFIS-based control model. The model was trained using a dataset derived from lateral and diagonal sensor values corresponding to various turning angles. The turning angle of the vehicle's own normal with the route varies from [−45°, +45°] according to distance data coming from sensors. Based on the sensor placement, when the vehicle moves parallel to the row, the distance M2 should be twice the distance M1.

D.1. ANFIS Training Stage

The data from the U1 and U2 sensors were utilized to create the training dataset for the ANFIS system. The HC-SR04 ultrasonic sensors used in this study have a detection range between 2 cm and 450 cm; however, a practical range of 0 to 80 cm was selected to ensure relevant measurements within the crop row. Various positions of the UGV within the row were recorded to generate a representative dataset [21].

Input 1 (M1) represents the lateral distance between the vehicle and the crop row, measured by the side ultrasonic sensors (U1 and U5). Input 2 (M2) denotes the diagonal distance, obtained from the diagonal sensors (U2 and U4). The output, referred to as the turning angle, determines how much the vehicle should rotate to stay aligned with the crop row. Based on M1 and M2, the ANFIS model computes the necessary angular correction, which is then sent to the UGV’s steering control system. The data from the lateral and diagonal sensors were used to develop an effective decision-making module. A training set between M1 and M2 is derived based on the minimum and maximum input ranges (inputs are decreasing from largest to smallest), with output values linearly mapped to turning angles between -45° and +45°, as to the relations defined in Table 5.

Table.5. Training Set M1 and M2 inputs and turning angle output [22].

M1(Cm)	M2(Cm)	Turning Angle (°)
20	80	+45
20	70	+37
20	60	+30
20	50	+22
20	40	…..
…..	….	…..

The ANFIS model was trained using this data. The ANFIS model was developed using the membership functions of the Sugeno type fuzzy inference system [21,22]. The model was created using the Neuro-Fuzzy Inference interface in Matlab and trained for 5000 epochs. After training, the error rate was observed to be 0.251. The model was tested with the weight file obtained from the training, and the FIS output versus Testing Data is presented in Figure 15.

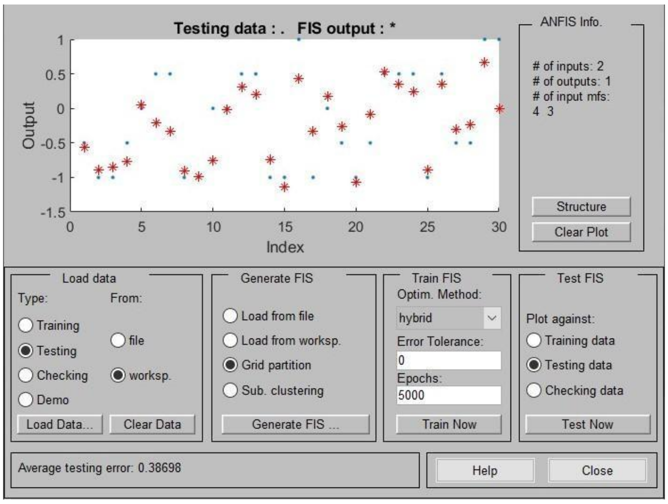

Fig.15. Test Data Results Corresponding to FIS Results

The surface plot resulting from the model training is shown in Fig.16. This plot, generated using lateral and diagonal sensor data, illustrates that the model produced proportionally consistent outputs across varying distances. The predicted turning angles for the vehicle were observed to remain within the range of -45° to +45°, as also supported in [23].

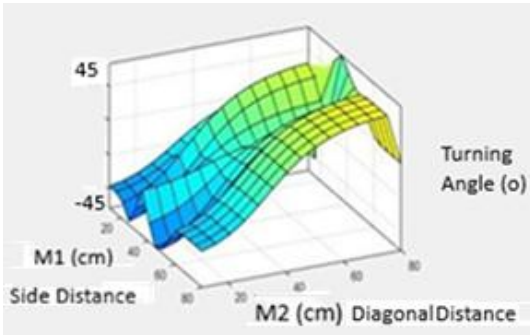

Fig.16. Surface Plot for 5000 Epochs

In the model training phase, the ‘gauss’ membership function (gaussmf) was preferred as the membership function [24]. Membership Functions for the “Side Distance Data” input variable and the “Diagonal Distance Data” input variable after 5000 training iterations are given in Fig.17 and Fig.18, respectively.

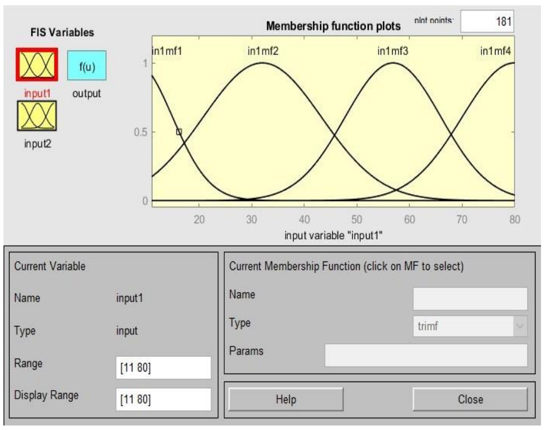

Fig.17. Membership Functions of “Side Sensor Distance Data”

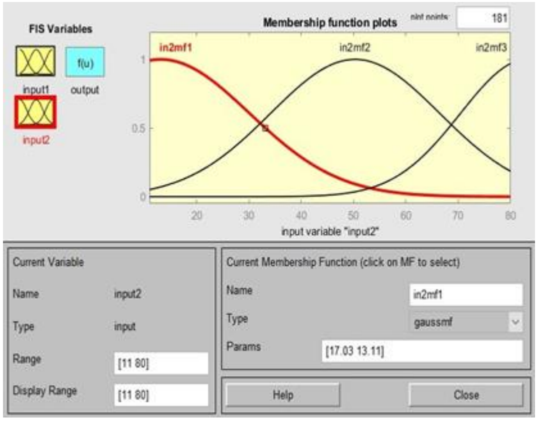

Fig.18. Membership Functions of "Cross Sensor Distance Data

D.2. ANFIS Testing Stage

After training, the ANFIS model was tested using unseen input data, and the predicted turning angles closely matched the expected values (Fig.19).

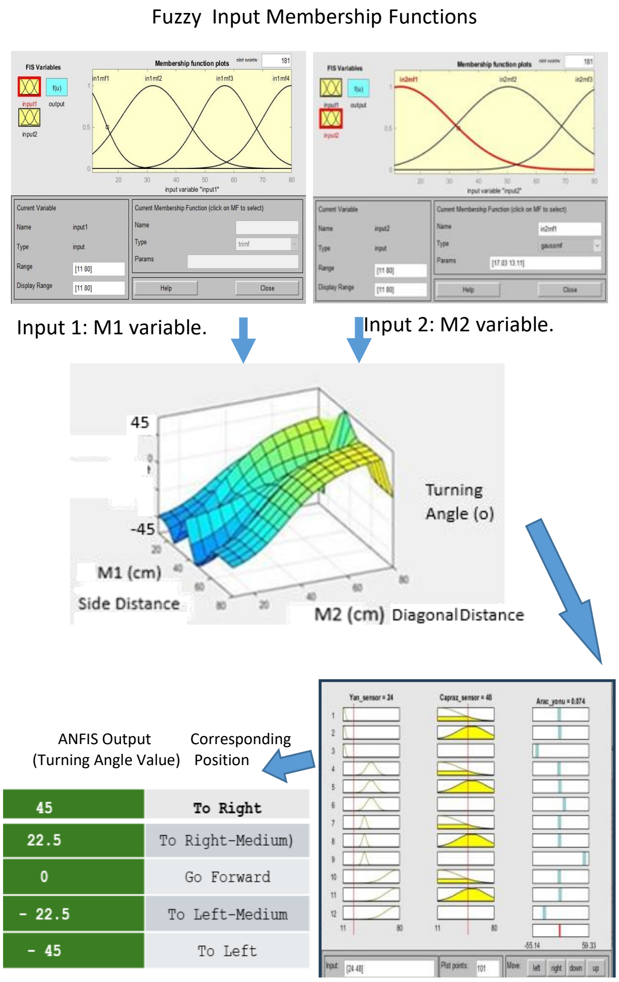

Fig.19. Testing stage of the trained ANFIS Model

The FIS outputs align well with the test data, confirming the model's accuracy. The resulting surface plot demonstrates that the relation surface correctly reflects the input distance data and corresponding turning angles within the range of -45° to +45°. The final ANFIS surface was then embedded into the Arduino Uno as a lookup table to enable real-time steering control of the UGV.

E. Driving – Control Module

The UGV is driven by four motors, each controlled via a dedicated motor driver managed by the Arduino Uno. A motor assignment matrix is used to determine the appropriate PWM speed signals for each motor based on the desired turning angle and movement direction (Table.6). This matrix translates the control commands from the ANFIS output into differential motor speeds, enabling smooth and accurate maneuvering of the vehicle within the crop rows (Fig.20).

Table.6. Motor Assigment Matrix

Vehicle Direction Speed (m/s)	**Right Motor Group (m/s)**	**Left Motor Group (m/s)**
Forward	Maximum Forward Speed	Maximum Forward Speed (PWM 255)
Right	0	Maximum Forward Speed
Right-medium	Maximum Forward Speed*(1/3)	Maximum Forward Speed*(2/3)
Right-small	0	Maximum Forward Speed*(1/3)
Left	Maximum Forward Speed	0
Left-medium	Maximum Forward Speed*(2/3)	Maximum Forward Speed*(1/3)
Left-small	Maximum Forward Speed*(1/3)	0
Backward	Maximum Backward Speed	Maximum Backward Speed
Backward-medium	Maximum Backward Speed*(2/3)	Maximum Backward Speed*(2/3)
Backward-small	Maximum Backward Speed*(1/3)	Maximum Backward Speed*(1/3)
STOP	0	

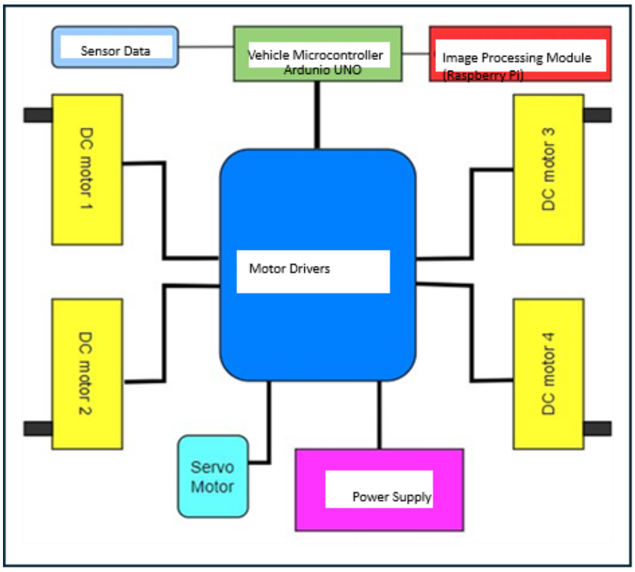

Fig.20. UGV Motor Control Block Scheme

RESULTS

Throughout the study, the goal was to autonomously detect and spray diseased plants in real-time using a UGV and the YOLOv8 model running on a Raspberry Pi mini computer.

In this purpose a crop field environment was simulated in Kocaeli University, Department of Electronics and

Communication Engineering, Marine Robotics Research Laboratory (DRAL/MARAS) [28]. This area is created by arranging identical photographs of diseased and healthy plants in a row (Figure 21).

Fig.21. Case Area

The UGV inferred and navigated along the crop row regulary at sufficient speed and injected medicine to the detected diseased plants (Fig.22).

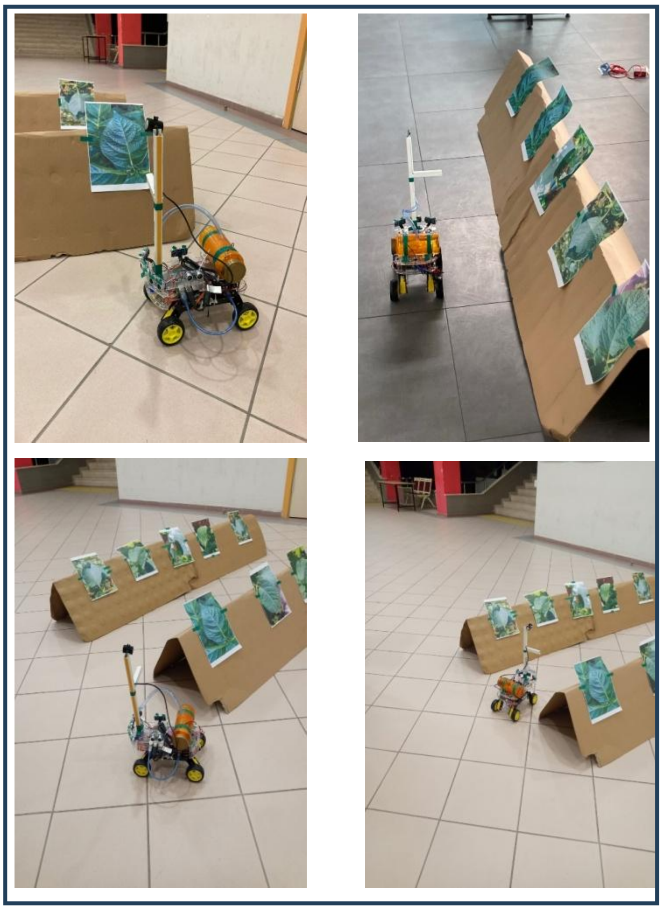

Fig.22. Navigation and Processing of UGV in the Test Area

Images of the study area and its outcomes are provided in Figures 23 and 24.

Fig.23. Detecting and Spraying Simulations by the UGV

The YoloV8 algorithm detected diseased leaves with high accuracy.

Fig. 24. YOLOv8 Detects Infected Leaves.

II. DISCUSSION

Studies on plant disease classification [25], autonomous agricultural robotic systems [26], and real-time monitoring technologies [27] contribute significantly to the advancement of agricultural automation and increases the overall yield percentage. In this study, during the test process, it was observed that UGV detected diseased potato leaves with high accuracy (average 79%). Following the detection, the spraying process was carried out using the spraying module on the vehicle. Since the 'diseased' information was delayed by 3.5-4 seconds when the disease detection process was carried out, it was actually solved with the algorithm designed on the main processor on computer.

The reported average detection accuracy of 79% was achieved in a real-time operational scenario, where the YOLOv8 model was deployed on a Raspberry Pi 4B. While this platform is energy-efficient and cost-effective, it introduces computational limitations that constrain model complexity and inference speed. The goal of this study was to implement a fully integrated, low-cost, real-time disease detection and autonomous spraying system under field-like conditions.

When benchmarked against similar studies, the result remains competitive. For example, Shill [1] reported that YOLOv3 and YOLOv4 yielded average accuracies of 53.08% and 55.45%, respectively, for multi-class plant disease detection. Verma et al. [6] achieved over 90% accuracy using classical machine learning methods, but their approach was offline, lacked real-time performance, and did not involve autonomous operation. Dung [7] implemented SSD (Single Shot Detector) on a Raspberry Pi 3 with ~72% accuracy, but without integrated motion control or spraying. Similarly, Kumar and Dineshraja [8] developed a Raspberry Pi-based rover with environmental sensing and navigation but reported a disease detection latency of 95 seconds per image and lack of spraying function.

Unlike these works, our system performs real-time detection, navigation, and targeted spraying in a fully integrated and autonomous loop. While the 79% accuracy may seem lower than some offline benchmarks, it is acceptable and meaningful within the context of embedded, real-time agricultural robotics. A delay of approximately 4 seconds was observed between detection and spraying due to onboard inference and control processes. This latency was mitigated by limiting the UGV's speed to 0.15–0.20 m/s, minimizing the spatial offset between detection and actuation. Additionally, the UGV incorporates a basic reactive behavior: it slows down or reverses briefly when diseased leaves are detected, allowing time for proper alignment and spraying. While this approach may limit scalability in larger plots, it demonstrates the feasibility of real-time, low-cost disease management in semi-structured environments. Future improvements will focus on latency reduction and system robustness through hardware upgrades and sensor fusion. Wheel speed drift introduces small but non-negligible targeting errors in our spraying system, especially under real-time constraints. While these are currently managed by operating at low speeds and re-adjustments refer to sensorial information, future versions of the platform will incorporate wheel encoders and closed-loop control to ensure consistent drive behavior and improve targeting accuracy during spraying.

III. CONCLUSION

In this study, an autonomous unmanned ground vehicle (UGV) was developed and tested for disease detection and precision spraying in agricultural environments. The system integrates image processing via the YOLOv8 algorithm and autonomous navigation using an ANFIS-based decision model. The integration of YOLOv8 and ANFIS with embedded hardware platforms contributes to both scientific innovation and to real-world applicability. Experimental results in a controlled test area demonstrated that the UGV could reliably follow crop rows, detect diseased plants, perform targeted spraying, and execute accurate turns. Despite minor issues such as sensorial insensitivities, processing delays on the Raspberry Pi, and wheel speed variations, the overall system achieved the desired performance. We conducted our experiments in a laboratory environment with standard, fixed lighting conditions. In these experiments, the vehicle detected diseased plants with an accuracy of 79% on images that we identified as diseased. Outside of this controlled setting, in a real field with plants and varying light conditions, both navigation and disease detection performance are expected to decrease.

The successful implementation of real-time disease detection and autonomous control highlights the system's potential for practical applications in precision agriculture. By minimizing unnecessary chemical usage and enabling efficient resource allocation, the proposed solution contributes to more sustainable and environmentally friendly farming practices.

ACKNOWLEDGMENT

Authors would like to thank Hafis Guliyev (Msc) for his valuable contributions and guidance.

Authors' Contributions

The authors' contributions to the paper are equal.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The authors declare that this study complies with Research and Publication Ethics

REFERENCES

[1] A. Shill, "Plant Disease Detection Based on YOLOv3 and YOLOv4," in Proc. Int. Conf. on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, Jul. 8–9, 2021.

[2] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," Information Processing in Agriculture, 2016.

[3] J. A. Soeb, F. Jubayer and I. Meftaul, "Tea leaf disease detection and identification based on YOLOv7," Scientific Reports, [Online].

[4] T.Y. Mahesh, "Leaf Disease Detection in Bell Pepper Using YOLOv5," International Journal of Engineering Research & Technology (IJERT), Vol. 12, 2023.

[5] M.P. Reddy and A. Deeksha, "Mulberry Leaf Disease Detection Using YOLO," International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 7, No. 3, 2021.

[6] S. Verma, D. Khare, R. Gupta and G. Singh, "Analysis of Image Segmentation Algorithms Using MATLAB," Proceedings of the Third International Conference on Trends in Information, Telecommunication and Computing, pp. 163-172, 2012.

[7] L. Dung, "Real-Time Tomato Leaf Disease Detection Using Single Shot Detector on Raspberry Pi 3," Unpublished.

[8] S. Kumar and K. Dineshraja, "Development of a Real-Time Plant Species Recognizing Rover," 13th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2021.

[9] R. Sharanesh, et al., "Plant Disease Detection in Strawberry and Grape Using VGGNet16 and InceptionResNetv2 Architectures," 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2019.

[10] J.O. Ahmed, et al., "Real-Time Agricultural Monitoring with Agrobot: A Raspberry Pi and YOLO Based Solution", International Conference on Computer and Applications (ICCA), Egypt, 28-30 November 2023.

[11] R. Gorapudi, B.R.S.P. Rudrapaka and A.S. Valluri, "Design and Implementation of Pesticide Spraying Robot Using IoT," International Journal of Advance Research and Innovation, Vol. 8, No. 2, pp. 36-41, 2020.

[12] M. S. Vani, S. Girinath, V. Hemasree, L. H. Havardhan, and P. Sandhya, "Plant disease identification tracking and forecasting using machine learning," in Proc. 2023 3rd Int. Conf. Technological Advancements in Computational Sciences (ICTACS), Nov. 1–3, 2023, pp. 1428–1432, Conf. ID: 59847

[13] R. Hamzah, L. Ang, R. Roslan, N. H. I. Teo, K. A. Samad, and K. A. F. A. Samah, "Comparing modified YOLO v5 and Faster Regional Convolutional Neural Network performance for recycle waste

classification," in Proc. 2024 IEEE Int. Conf. Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 29 June 2024, pp. 415–419.

[14] S. Pandey, K.-F. Chen and E. B. Dam, "Comprehensive Multimodal Segmentation in Medical Imaging: Combining YOLOv8 with SAM and HQ-SAM Models," 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2023, pp.2584-2590, doi: 10.1109/ICCVW60793.2023.00273.

[15] S. Pandey, K.-F. Chen and E. B. Dam, "Comprehensive Multimodal Segmentation in Medical Imaging: Combining YOLOv8 with SAM and HQ-SAM Models," 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2023, pp.2584-2590, doi: 10.1109/ICCVW60793.2023.00273.

[16] P. Sajitha, A. John, V. L. Devika, S. V. Gayathri and N. Sakhir, "Leaf disease detection & correction using YOLOv7 with GPT-3 integrated," Int. J. of Engineering Research & Technology (IJERT), vol. 12, 2023.

[17] H. Yi, B. Liu, B. Zhao ve E. Liu, "Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol.17, pp. 1734–1747, 2024

[18] S. Yılmaz And S. B. Kilci, "Modeling and simulation of a fuzzy heat distribution controlled high-voltage DC resistive divider," MEASUREMENT & CONTROL , vol.53, pp.485-500, 2020

[19] S. Yılmaz Et Al. , "Erosion rate of AA6082-T6 aluminum alloy subjected to erosive wear determined by the meta-heuristic (SCA) based ANFIS method," MATERIALS TESTING , vol.66, no.2, pp.248-261, 2024

[20] O. Rachidi, E. Chafik and B. Bououlid, "Design of a real-time-integrated system based on stereovision and YOLOv5 to detect objects," [Online].

[21] H. Guliyev and S. Yılmaz, "REAL-TIME DISEASE DETECTION USING THE YOLO ALGORITHM WITH A MODULE MOUNTED ON A ROVER," 6th International Azerbaijan Congress on Life, Engineering, Mathematical, and Applied Sciences , vol.1, no.1, Baku, Azerbaijan, pp.247-254, 2024

[22] H.Guliyev (Student), Postgraduate,2024, Modeling of Agricultural Unmanned Ground Vehicle and Motion Control Applications Using ANFIS Methods, M.Sc. Thesis, S. Yılmaz (Advisor), Kocaeli University Institute of Science, Kocaeli

[23] K. Dineshraja and S. Kumar, "Development of a real-time plant species recognizing rover," in Proc. 13th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), 2021.

[24] H. T.Abatari, and A. D. Tafti, "Using a fuzzy PID controller for the path following of a car-like mobile robot". First RSI/ISM IEEE international conference on robotics and mechatronics (ICRoM) (pp. 189-193), February 2013.

[25] P. Panchal, V. C. Raman and S. Mantri, "Plant diseases detection and classification using machine learning models," in Proc. 4th Int. Conf. on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2019.

[26] R. Abhiram and R. K. Megalingam, "Autonomous Fertilizer Spraying Mobile Robot," in Proceedings of the 2022 IEEE 19th India Council International Conference (INDICON), Bengaluru, India, Nov. 2022, pp. 1–6.

[27] A. J. Ahmed, A. Babiker, A. Elhag, and M. Drar, "Real-Time Agricultural Monitoring with Agrobot: A Raspberry Pi and YOLO Based Solution," in Proceedings of the 2023 International Conference on Computer and Applications (ICCA), Kuala Lumpur, Malaysia, Nov. 2023, pp. 1–5.

[28] https://avesis.kocaeli.edu.tr/arastirma-grubu/dral