Smart Farming System

Alka Denny
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
alkaden2020@gmail.com

Anithra Ross Ajith
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
anithrarossajith1@gmail.com

Anna Joju
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
jojuanna12@gmail.com

Anu Xavier
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
anuannaxavier73@gmail.com

S. Santhi Jabarani
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
santhij@rajagiritech.edu.in

Swapna Davies
Dept. of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, India
swapnad@rajagiritech.edu.in

***Abstract*—Agriculture faces significant challenges due to plant diseases and inefficient water usage, leading to reduced crop yield and resource wastage. This paper presents a smart agriculture system integrating deep learning and IoT technologies to enhance precision in disease detection and irrigation. A ResNet50-based Convolutional Neural Network (CNN) model is employed for accurate leaf disease classification, enabling targeted fertilizer spraying based on the identified disease. Additionally, a soil moisture sensor ensures precise irrigation, conserving water by supplying it only when necessary. The system also incorporates real-time weather monitoring to provide farmers with environmental insights for better decision-making. By leveraging automation and machine learning, this approach enhances crop health, optimizes water usage, and improves agricultural productivity. Experimental results demonstrate the system's effectiveness in reducing resource consumption while maintaining high accuracy in disease detection.**

Keywords—Classification, Machine learning, Precise irrigation, Agriculture

I. Introduction

Agriculture plays a vital role in the global economy, with a significant portion of the population depending on it for livelihood. However, traditional farming practices often result in challenges such as low productivity and inefficient resource utilization. One of the major threats to crop yield is plant diseases, which can cause significant economic losses if not detected and managed promptly. The emergence of deep learning techniques, particularly Convolutional Neural Networks (CNNs), has shown promise in addressing these challenges by enabling accurate and automated detection of plant diseases through image-based analysis [1].

In recent years, ResNet50, a deep residual learning framework, has gained prominence for its ability to handle complex image classification tasks with high accuracy. Its use of residual blocks helps mitigate the vanishing gradient problem, allowing the model to learn deeper representations effectively. Studies have demonstrated the efficacy of ResNet50 in identifying various plant diseases with remarkable accuracy. For instance, a study employing ResNet50 for detecting rice plant diseases achieved an accuracy of 96.35%, highlighting its potential for real-world agricultural applications [2].

Building on this foundation, the proposed system integrates ResNet50 for leaf disease detection with a precise irrigation and fertilizer management strategy. The system uses a Raspberry Pi 4 as the processing unit to execute the model and manage data from various sensors. Soil moisture sensors provide real-time data to control irrigation levels precisely, ensuring optimal water usage based on soil conditions. Simultaneously, the disease classification results from ResNet50 inform the fertilizer management system, enabling the application of appropriate nutrients to mitigate the impact of identified diseases [1].

The adoption of advanced computing techniques in agriculture is transforming traditional practices. Devices such as soil moisture sensors, combined with powerful processors like Raspberry Pi 4, facilitate real-time monitoring and control of farming activities. The integration of these technologies not only enhances crop health monitoring but also contributes to resource conservation by preventing excessive water and fertilizer use [1], [2].

The proposed system's architecture consists of three main modules: disease detection, irrigation control, and fertilizer management. The disease detection module captures and preprocesses leaf images, which are then fed into the ResNet50 model for classification. Based on the detected disease, the system suggests specific fertilizers to address nutrient deficiencies caused by the disease. The irrigation control module utilizes data from soil moisture sensors to adjust water levels precisely, reducing waste and ensuring that crops receive adequate hydration. The entire process is managed by the Raspberry Pi 4, which coordinates sensor data collection, image processing, and actuator control [1].

The use of Raspberry Pi 4 is particularly advantageous due to its low cost, energy efficiency, and compatibility with machine learning frameworks like TensorFlow and PyTorch. It serves as a bridge between sensors and the deep learning model, enabling efficient execution of tasks in a resource-constrained environment. Moreover, the proposed system's reliance on open-source software and affordable hardware makes it accessible to small-scale farmers, promoting the adoption of smart farming practices in developing regions [2].

In conclusion, the integration of ResNet50 for leaf disease detection with precise irrigation and fertilizer management presents a comprehensive solution for enhancing agricultural productivity. By leveraging deep learning and edge computing on the Raspberry Pi 4 platform, the system addresses critical challenges in agriculture, including disease management, water conservation, and optimal fertilizer usage. Future work will focus on expanding the model's capabilities to detect a broader range of diseases and incorporating additional sensors for enhanced monitoring [1], [2].

II. RELATED WORKS

Recent advancements in smart agriculture have focused on integrating IoT, artificial intelligence, and automation to enhance farming efficiency. Various studies have explored automated irrigation, disease detection, and robotic systems, highlighting the transformative impact of technology in modern agriculture.

In [1], an IoT-enabled agricultural robot is presented for smart cultivation. This system employs multiple sensors to monitor real-time soil and environmental conditions, ensuring precise water and fertilizer distribution. The research emphasizes the benefits of data-driven decision-making for improved crop management.

The study in [2] introduces an automated irrigation system utilizing wireless sensor networks. The system leverages soil moisture sensors to regulate water supply dynamically, reducing wastage and improving irrigation efficiency. The findings indicate that automation significantly enhances water resource management in agriculture. Deep learning-based plant disease detection has gained prominence in recent research. In [3], a hybrid method for palm leaf disease detection is proposed, integrating image processing with machine learning techniques. The approach facilitates early disease identification, preventing potential crop losses and improving yield outcomes.

Furthermore, [4] explores plant disease detection using ResNet50, a deep learning model renowned for image classification. The authors illustrate how convolutional neural networks (CNNs) significantly boost disease identification accuracy, enabling timely and precise agricultural interventions.Another significant work in [5] discusses the integration of automated agricultural mechanisms with relay modules for precise actuation. The study presents an intelligent spraying system that activates upon disease detection, ensuring targeted pesticide and fertilizer application while minimizing resource wastage.

Collectively, these studies underscore the vital role of automation, sensor networks, and AI-driven disease detection in smart farming. The convergence of these technologies paves the way for a more sustainable and efficient agricultural ecosystem, serving as the foundation for our proposed system.

III. SYSTEM MODEL AND METHODS

The smart farming system presented in this study integrates deep learning and automation to enhance agricultural efficiency by enabling real-time disease detection, soil moisture assessment, and precision-based interventions. The system is designed to function autonomously, reducing human dependency while improving the accuracy and effectiveness of farming practices. At its core, the system consists of a mobile rover that follows a predefined track, scanning plant leaves for potential diseases using a deep learning model. Upon detecting a disease, the system initiates a sequence of automated actions, including soil moisture measurement, water sprinkling, and fertilizer application.

To achieve accurate disease classification, the system employs a dual-model deep learning approach, leveraging DenseNet-121 and ResNet-50. The datasets used for training and evaluating the plant disease detection model were collected from publicly available sources such as Kaggle and Mendeley. These datasets comprise a diverse range of leaf images from crops like tomato, chili, spinach, and others, covering both healthy and diseased samples. Each image in the dataset is labeled with its corresponding class, enabling supervised learning for disease classification. The inclusion of multiple crop types and various disease categories ensures that the model is trained on a wide variety of patterns and symptoms, enhancing its robustness and accuracy in real-world farming environments. These rich and well-curated datasets played a crucial role in building a reliable and effective deep learning model for smart agriculture applications.

The rover is equipped with a high-resolution camera for continuous image acquisition, a motor driver for controlled movement, and an intelligent stopping mechanism that halts operations when a diseased plant is detected. This allows for precise application of treatments and targeted analysis of affected areas. The system also features an automated irrigation mechanism, triggered based on real-time soil moisture readings. If the soil is dry, water sprinkling is activated to optimize hydration. Similarly, a fertilizer spraying mechanism ensures timely treatment for diseased plants. Additionally, continuous environmental monitoring is integrated through sensors that measure temperature and humidity every two seconds, providing valuable insights into farm conditions. By combining machine learning with automation, this methodology ensures resource efficiency, minimizes wastage, and enhances disease detection accuracy. The system's real-time processing capabilities and autonomous decision-making offer a scalable and innovative solution for modern agriculture, paving the way for improved crop health management and sustainable farming practices.

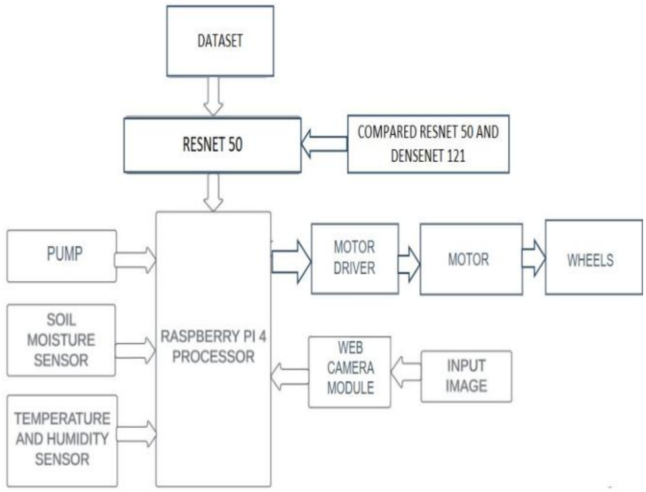


Fig 1: Block Diagram of the Smart Farming System

A. Leaf Disease classification

The smart farming system employs deep learning-based classification to detect leaf diseases with high accuracy. Two advanced convolutional neural networks (CNNs), DenseNet-121 and ResNet-50, are used to analyze plant leaf images and classify them as either healthy or affected by diseases. These models were chosen for their strong feature extraction capabilities, which enhance the precision of disease detection.

a. Densenet121 model

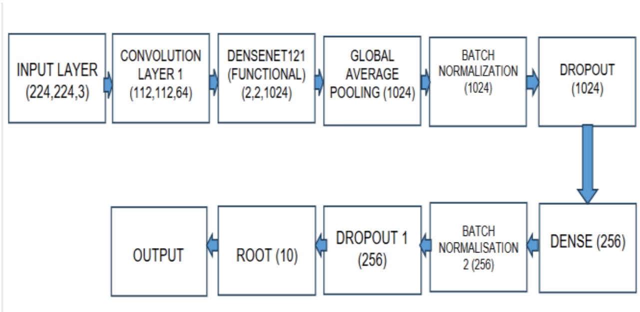


Fig 2: DenseNet121 Architecture

Fig 2 utilizes the DenseNet121 model as a feature extractor within a custom deep learning classification pipeline. The model begins with an input layer that accepts RGB images of size 224x224x3, followed by an initial convolutional layer that reduces the spatial dimensions and extracts low-level features. The output is then passed into a pre-trained DenseNet121 model (excluding its top classification layers), which outputs a rich feature map of size 2x2x1024. This output undergoes Global Average Pooling to flatten it into a 1024-dimensional feature vector. To enhance learning stability and prevent overfitting, Batch Normalization and Dropout layers are applied. The resulting features are further passed through a dense (fully connected) layer with 256 neurons, followed by another batch normalization and dropout layer. Finally, the network ends with a dense layer named "Root" with 10 units, which likely corresponds to the number of target classes. The final output layer provides the predicted class probabilities. This structure leverages DenseNet121's robust feature extraction capabilities while adapting the final layers for a specific classification task through transfer learning.

b. ResNet50 model

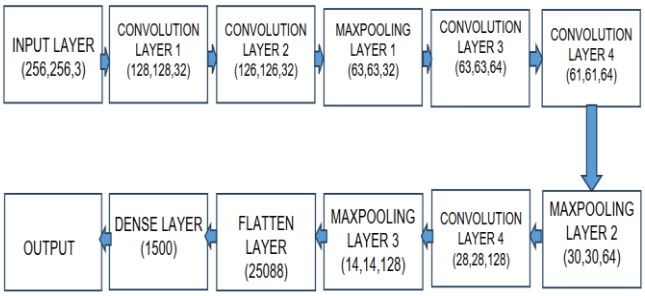


Fig 3: ResNet50 Architecture

Fig 3 represents a custom convolutional neural network (CNN) inspired by ResNet50 principles but does not directly depict the standard ResNet50 structure. It starts with an input layer for RGB images of size 256x256x3, followed by two convolutional layers that progressively extract low-level features, reducing the spatial dimensions. MaxPooling is applied to downsample the feature maps, after which deeper convolutional layers (with increased filter depth) further refine feature extraction. Another MaxPooling layer reduces the spatial size, followed by two more convolutional layers with 128 filters, extracting more complex patterns. The final MaxPooling operation downsamples the feature maps to 14x14x128. These are then flattened into a 25,088-dimensional vector and passed through a dense layer with 1500 neurons. The model ends with an output layer, likely used for classification. Though titled ResNet50, this architecture does not include residual connections, which are the hallmark of ResNet. Instead, it appears to be a simplified, custom CNN influenced by ResNet's deep structure design philosophy.

In our project, we chose the ResNet-50 model over DenseNet-121 primarily due to performance limitations observed during real-time implementation on the Raspberry Pi 4. While both models delivered good accuracy, DenseNet-121 caused significant delays when processing real-time images from the web camera module. This lag hindered the responsiveness required for timely disease detection in smart farming applications. In contrast, ResNet-50 offered a more efficient balance between accuracy and inference speed, making it better suited for deployment on resource-constrained devices like the Raspberry Pi. Its faster processing enabled smoother real-time performance, ensuring prompt and reliable detection in field conditions.

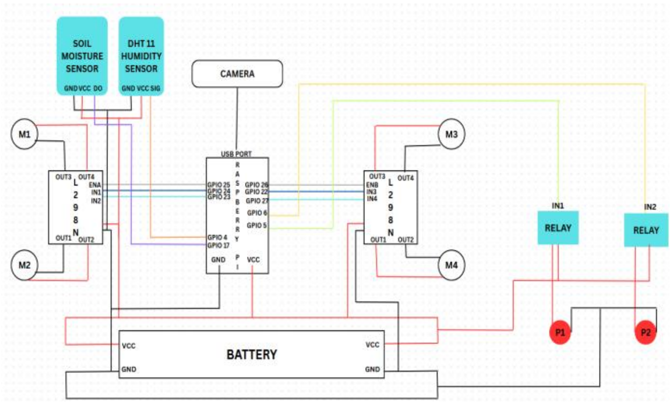


Fig 4: Circuit Diagram of the Smart Farming System

B. Rover Navigation and Automated Stopping Mechanism

The rover is a vital part of the smart farming system, designed to autonomously navigate a predefined track while scanning crops for diseases. Controlled by L298N motor driver modules, the rover uses four DC motors (M1–M4) to drive its wheels, enabling systematic movement across the field. A web camera module connected via the USB port to the Raspberry Pi 4 captures real-time images of the crops. These images are processed by the ResNet-50 deep learning model, which runs efficiently on the Raspberry Pi due to its balance between performance and computational load, making it ideal for embedded systems.

A key feature of this system is the automated stopping mechanism, triggered when the ResNet-50 model detects a diseased plant. Upon detection, the rover halts, and a soil moisture sensor checks the moisture level in the soil. If the soil is dry, the Raspberry Pi activates a relay module, which controls Pump 1 (P1) to sprinkle water. Simultaneously, another relay controls Pump 2 (P2) to spray fertilizer on the diseased plant. A DHT11 temperature and humidity sensor is also connected to monitor environmental conditions, which are displayed or used for data logging. The entire system is powered by a battery pack that ensures portability and uninterrupted field operation. All sensors and actuators are connected through appropriate GPIO pins of the Raspberry Pi, chosen for their availability and ease of control in Python. This integrated use of components ensures real-time decision-making, efficient resource utilization, and automation in precision agriculture, reducing manual labor and increasing productivity.

C. Soil Moisture Detection and Water Sprinkling System

The smart farming system includes a soil moisture detection module to ensure efficient irrigation. A soil moisture sensor connected to the Raspberry Pi 4 GPIO pins is used to measure the water content in the soil specifically at the location where a disease is detected by the system. This sensor was chosen for its low power consumption, simple interfacing, and real-time analog output, making it ideal for embedded applications like ours. If the sensor detects dry soil, the Raspberry Pi processes this information and triggers a relay module connected to Pump 1 (P1), which activates the water sprinkling mechanism. The relay acts as a switch that safely controls the high-power pump from the low-power GPIO output of the Raspberry Pi. This setup ensures water is used only when necessary, preventing overwatering and conserving valuable resources.

The decision-making process is entirely automated, relying on real-time sensor readings and the microcontroller's logic. By integrating these components—soil moisture sensor, Raspberry Pi, relay module, and water pump—the system effectively optimizes irrigation management. This not only improves crop health by maintaining ideal soil moisture levels but also promotes sustainable water usage in agriculture, aligning with the goals of precision farming.

D. Fertilizer Spraying for Disease Treatment

The smart farming system integrates an automated fertilizer spraying mechanism to ensure timely and targeted treatment of diseased crops. Upon disease detection by the ResNet-50 model, a control signal is sent from the Raspberry Pi 4 to Relay Module 2, which activates Pump 2 (P2) connected to the fertilizer sprayer. The use of a relay module allows the system to safely switch high-power components like pumps using low-power GPIO signals. This design ensures that fertilizer or pesticide is applied only when required, improving accuracy and reducing excessive chemical usage. The controlled and responsive mechanism minimizes manual effort, optimizes chemical consumption, and supports sustainable farming by ensuring precise intervention at the right time.

E. Continuous Environmental Monitoring

The smart farming system features continuous environmental monitoring to track key parameters like temperature and humidity in real time. These critical factors influence plant health and disease vulnerability. A DHT11 temperature and humidity sensor is used to collect environmental data at regular two-second intervals. This sensor was chosen for its simplicity, reliability, and compatibility with the Raspberry Pi 4 GPIO pins, making it ideal for low-cost and efficient climate sensing. The real-time data gathered is processed by the Raspberry Pi 4, allowing immediate insights into the farm's microclimate. This information supports informed decisions for disease prevention and optimized irrigation. By automating climate monitoring, the system minimizes manual intervention, ensures optimal growing conditions, and enhances overall crop productivity. The integration of the DHT11 sensor ensures proactive responses to environmental changes, making farming more efficient and sustainable.

V. RESULTS AND DISCUSSION

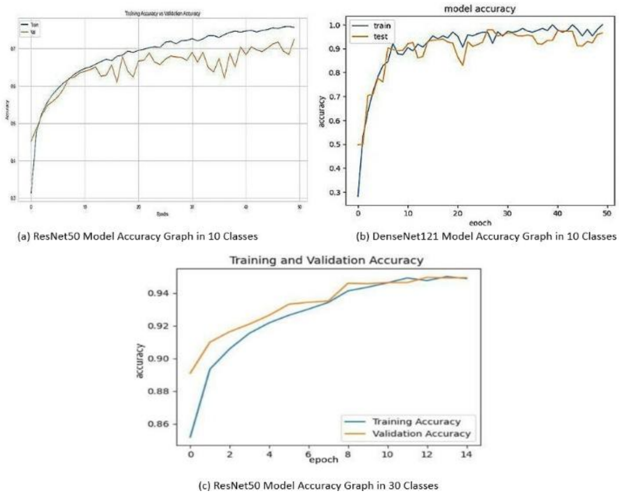
This section presents the outcomes of implementing the smart farming system, highlighting the performance of disease detection, automation accuracy, and real-time responsiveness. It also discusses the effectiveness of the chosen components and models in achieving the project's objectives.

A. Model Accuracy and Loss evaluation

a. DenseNet121 vs Resnet50 Model Accuracy Curve

The three graphs presented illustrate the training and validation accuracy trends of two deep learning models—ResNet50 and DenseNet121—under different class settings.

Each graph provides insight into how the models perform in terms of learning capacity, generalization, and stability, helping guide the decision for the best model to use in the final implementation.



(a) ResNet50 Model Accuracy Graph in 10 Classes (b) DenseNet121 Model Accuracy Graph in 10 Classes

(c) ResNet50 Model Accuracy Graph in 30 Classes

Fig 4: Accuracy of training and validation data

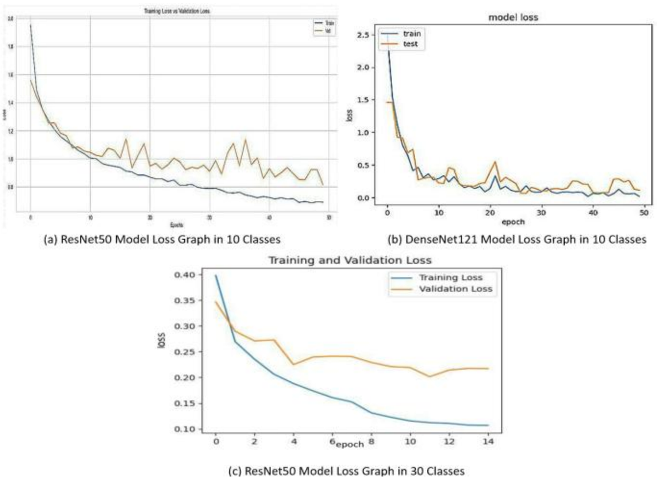
The first graph (a) shows the accuracy curve of the ResNet50 model trained on a dataset with 10 classes. Both training and validation accuracy rise steadily across the epochs. The x-axis represents the number of epochs, while the y-axis denotes the accuracy. Training accuracy surpasses 90% and continues to improve gradually, while validation accuracy also trends upward with minor fluctuations. The close alignment between training and validation lines suggests that the model is learning effectively without significant overfitting. This indicates that ResNet50 generalizes well on unseen data while maintaining high accuracy, making it a reliable choice for classification tasks involving 10 categories.

In contrast, graph (b) displays the DenseNet121 model accuracy when trained on the same 10-class dataset. The model shows a rapid increase in both training and test accuracy during the early epochs, quickly reaching over 90%. However, the validation accuracy fluctuates more noticeably than ResNet50, especially around the 20–30 epoch range. These inconsistencies imply that DenseNet121, while powerful, may be more sensitive to the training dynamics, and could be prone to minor overfitting or instability in validation performance. Though it performs well in terms of raw accuracy, this lack of consistency might be a limitation in real-world applications. The third graph (c) presents the ResNet50 model accuracy on a more complex dataset with 30 classes. Remarkably, the model still performs strongly, with training and validation accuracy approaching 94%. The curves are smooth and closely follow each other throughout the training process, showing that ResNet50 scales well to increased class complexity without sacrificing performance. The very small gap between training and validation accuracy further supports its excellent generalization capability.

Based on these observations, ResNet50 was chosen over DenseNet121 for the final implementation. ResNet50 offers smoother and more stable learning curves, demonstrating consistent performance even when the number of classes increases. It generalizes well, showing little to no overfitting across different tasks. Furthermore, ResNet50 is relatively lightweight compared to DenseNet121, making it more suitable for real-time applications or deployment on resource-constrained devices like the Raspberry Pi. While DenseNet121 is also a strong model, its slightly unstable validation performance and higher computational demands made ResNet50 the more practical and robust choice for the system.

b. Densenet121 vs Resnet50 Model Loss Curve

The graphs presented above depict the training and validation loss curves for the ResNet50 and DenseNet121 models, each trained on datasets with 10 and 30 classes respectively. These loss curves are essential for evaluating how well a model learns from data, and they also offer deeper insight into the model's generalization ability and risk of overfitting.



(a) ResNet50 Model Loss Graph in 10 Classes (b) DenseNet121 Model Loss Graph in 10 Classes

(c) ResNet50 Model Loss Graph in 30 Classes

Fig 5: Loss of training and validation data

The first graph (a) shows the loss curve of the ResNet50 model trained on a 10-class dataset. The training loss decreases steadily and significantly over the epochs, indicating that the model is effectively minimizing error on the training data. Although the validation loss follows a similar downward trend, it is more erratic, with frequent fluctuations. This variance may be attributed to either noise in the validation data or sensitivity to certain samples, but overall, the validation loss still trends downward. Importantly, the gap between training and validation loss is not extreme, suggesting that ResNet50 maintains reasonable generalization on unseen data. Graph (b) displays the DenseNet121 loss curve on the same 10-class dataset. Both training and validation losses start high and then rapidly decrease within the first few epochs, which is typical of deep learning models. The validation loss closely follows the training loss throughout the entire training period, indicating tight coupling between the two curves. However, small spikes are visible at several points in the validation curve, hinting at potential overfitting or unstable learning behavior.

Despite this, the average loss remains low, and the model performs well. Still, the instability observed during training could pose reliability concerns in deployment environments. The third graph (c) illustrates the loss behavior of ResNet50 on a more complex dataset with 30 classes. Here, both training and validation losses decrease consistently, with the training loss reaching very low levels. While the validation loss flattens and slightly fluctuates after epoch 6, it still continues its downward trend overall. The gap between the training and validation losses increases slightly as training progresses, which is expected in more complex classification tasks. Nevertheless, there is no sharp rise in validation loss, indicating that overfitting is well-controlled and the model remains stable across the training period.

Based on these loss graphs, ResNet50 again proves to be the better choice compared to DenseNet121. It demonstrates smooth and steady reduction in training loss and shows reliable validation loss trends even when the number of classes increases. DenseNet121, while showing good initial performance, exhibits minor instability in validation loss that could translate to inconsistent results in real-time applications. Additionally, ResNet50 appears more robust and adaptable to datasets of increasing complexity, as shown in the 30-class loss curve. These results further support the decision to select ResNet50 for the final implementation, balancing high accuracy with stable and predictable training behavior.

B. Farmbot

Fig 6: Structure of the Farmbot

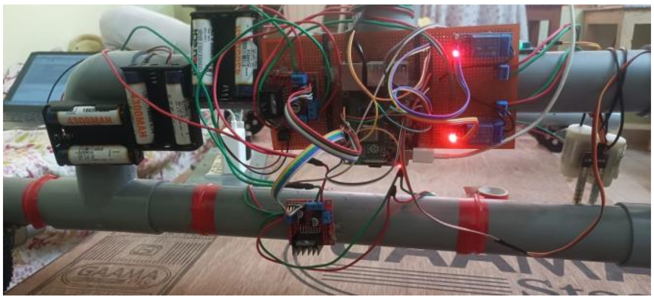
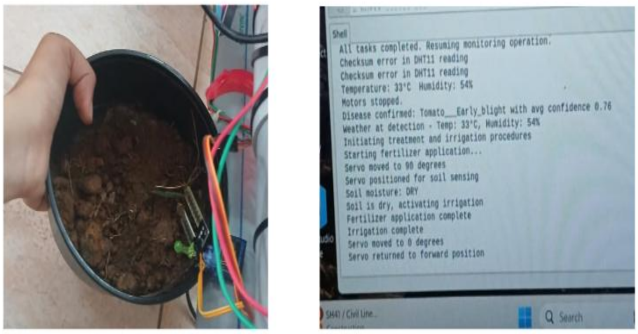


Fig 7: Connections To Raspberry Pi 4

Fig 6 displays the physical structure of the farmbot, constructed using PVC pipes to form a sturdy rectangular frame mounted on wheels for mobility. At the center, a camera module is attached to capture real-time images of crops for disease detection. The simple yet functional design allows the farmbot to navigate along predefined tracks in the field. Fig 7 presents the intricate circuitry and wiring connected to the Raspberry Pi processor. This includes multiple components such as battery packs for power supply, motor drivers, relay modules, and various sensors. The connections are mounted on a custom PCB board, demonstrating a compact and efficient hardware setup to support autonomous operations like image processing, soil moisture sensing, and actuation of sprinklers or fertilizer sprayers. Together, these images illustrate the blend of mechanical design and embedded electronics driving the functionality of the smart farming system.



(a) Dry Soil (b)Soil Moisture detecting the soil as dry

Fig 8: The soil moisture sensor detecting the soil as dry

Fig 8 illustrates the dry soil detection and treatment process in the smart farming system. In sub-image (a), a soil moisture sensor is inserted into a plant pot containing dry soil. This sensor is connected to the automated system powered by a Raspberry Pi. Sub-image (b) shows the terminal output from the Raspberry Pi, displaying the detection and decision-making process in real time.

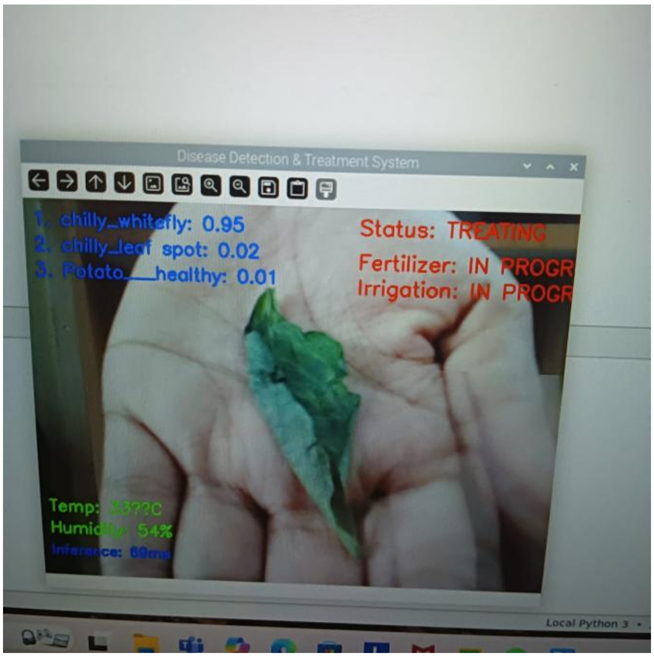


Fig 9: The output from the Smart Farming System

Fig 9 showcases the real-time output of the plant disease detection and treatment module. On the top-left, in blue text, the system displays the prediction results from the machine learning model: “chilly_whitefly” with a confidence of 95%, followed by “chilly_leaf spot” (2%), and “Potato_healthy” (1%). This clearly indicates that the leaf being analyzed is most likely infected by the whitefly pest. On the right side, in red, the Status is marked as “TREATING”, while both “Fertilizer” and “Irrigation” are indicated as “IN PROGRESS”, signifying that the system has autonomously initiated both processes. These operations are controlled based on real-time sensor readings, also displayed on the screen. At the bottom-left corner, in green, the system shows the current temperature (33°C) and humidity level (54%), captured through environmental sensors. The inference time is shown as 89 milliseconds, reflecting the model's efficiency in detecting disease. All of this data including prediction results, sensor values, and system statuses is transmitted wirelessly via Wi-Fi from the Raspberry Pi, which acts as the central processing unit of the smart farming system. Once the irrigation and fertilizer spraying tasks are completed, the respective labels will update from “IN PROGRESS” to “COMPLETED”, indicating the end of the treatment cycle.

VI. Conclusion

The smart farmbot developed in this project successfully automates essential farming tasks such as disease detection, irrigation, and fertilizer application. By autonomously moving across the farm, detecting leaf diseases, and responding with targeted fertilizer spraying, the system reduces manual labor and optimizes resource usage. Additionally, it monitors soil moisture levels and activates irrigation when necessary, ensuring efficient water management. Despite its success, the project faced challenges such as maintaining accurate disease detection under varying environmental conditions, optimizing the movement of the farmbot, and managing power consumption for continuous operation. However, the farmbot demonstrates a scalable and efficient approach to modernizing agriculture through automation, helping farmers improve productivity while minimizing resource wastage.

Future advancements can further enhance the Farmbot’s capabilities and sustainability. A moving camera module can be integrated to scan larger areas for better disease detection, while a weed removal mechanism using robotic arms can help eliminate unwanted plants, improving crop yield. AI-driven pesticide prevention can optimize chemical usage by applying pesticides only when necessary. Additionally, integrating solar panels will make the system energy-efficient, reducing dependency on external power sources. LoRa-based wireless communication can enable real-time tracking of soil conditions and farmbot performance over large farmlands. Further improvements, such as AI-driven decision support for predictive farming and multi-crop support, will make the system more adaptable and intelligent. With these advancements, the farmbot has the potential to revolutionize precision agriculture, making farming more automated, sustainable, and resource-efficient.

References

[1] Senthil Kumar Swami Durai, Mary Divya Shamili, Smart farming using Machine Learning and Deep Learning techniques, Decision Analytics Journal, Volume 3, 2022,100041,ISSN 2772-6622, https://doi.org/10.1016/j.dajour.2022.100041.

[2] U. Archana, A. Khan, A. Sudarshanam, C. Sathya, A. K. Koshariya and R. Krishnamoorthy, "Plant Disease Detection using ResNet," *2023 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal, 2023, pp. 614-618, doi: 10.1109/ICICT57646.2023.10133938.

[3] S. M. N. Nobel *et al*., "Palm Leaf Health Management: A Hybrid Approach for Automated Disease Detection and Therapy Enhancement," in *IEEE Access*, vol. 12, pp. 9097-9111, 2024, doi: 10.1109/ACCESS.2024.3351912.

[4] A. Salam, M. Naznine, N. Jahan, E. Nahid, M. Nahiduzzaman and M. E. H. Chowdhury, "Mulberry Leaf Disease Detection Using CNN-Based Smart Android Application," in *IEEE Access*, vol. 12, pp. 83575-83588, 2024, doi: 10.1109/ACCESS.2024.3407153.

[5] R. Rani, J. Sahoo, S. Bellamkonda, S. Kumar and S. K. Pippal, "Role of Artificial Intelligence in Agriculture: An Analysis and Advancements With Focus on Plant Diseases," in *IEEE Access*, vol. 11, pp. 137999-138019, 2023, doi: 10.1109/ACCESS.2023.3339375.

[6] M. S. Chaitanya, U. K. Reddy B, S. K and S. Amaran, "MFC Based Smart Irrigation System Using IoT and Machine Learning," *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, Goathgaun, Nepal, 2025, pp. 368-373, doi: 10.1109/ICMCSI64620.2025.10883570.

[7] J. Gutiérrez, J. F. Villa-Medina, A. Nieto-Garibay and M. Á. Porta-Gándara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," in *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 166-176, Jan. 2014, doi: 10.1109/TIM.2013.2276487

[8] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie and T. Shongwe, "Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey," in *IEEE Access*, vol. 11, pp. 145813-145852, 2023, doi: 10.1109/ACCESS.2023.3346299.

[9] E. -T. Bouali, M. R. Abid, E. -M. Boufounas, T. A. Hamed and D. Benhaddou, "Renewable Energy Integration Into Cloud & IoT-Based Smart Agriculture," in *IEEE Access*, vol. 10, pp. 1175-1191, 2022, doi: 10.1109/ACCESS.2021.3138160.

[10] S. Kar, R. Mohammad, M. T. Zaman, S. M. Reza, O. Talukder and M. Hasan, "Farmbot: an IoT-Based Wireless Agricultural Robot for Smart Cultivation," *2023 26th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh, 2023, pp. 1-6, doi: 10.1109/ICCIT60459.2023.10441291.