International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Plant Disease Detection Using Esp-32 With Machine Learning Model

Harshith S[1], Nikhilesh G[2], Raghunandan S[3], Shashank D S[4], Mrs. Shwetha S K[5]
[1,2,3,4]P.E.S. COLLEGE OF ENGINEERING, Mandya
Department of Electronics and Communication Engineering
[5]Assistant Professor, Dept. of ECE, PESCE, Mandya

Abstract- Crop illnesses remain one of the major sources of farm losses worldwide, and identifying them at an early stage can greatly improve yield protection. Many farmers still depend on manually walking across their fields and inspecting each plant, a process that consumes significant time and can be inconsistent. This study presents a low-cost detection system we developed using an ESP32-CAM to capture images of leaves and transmit them wirelessly to a cloud-based machine learning model. The system analyses each image to determine if the leaf is healthy or affected by a particular disease, and the outcome appears immediately on a web interface accessible to farmers via their phones. Our aim was to design an affordable and easy-to-use solution so that even smallholder farmers without technical expertise can operate it with ease and confidence in their daily farming activities without needing additional support.

Keywords- Medical image diagnosis, Frequency Feature, Clustering, DIP.

I. INTRODUCTION

India is among the world's largest agricultural nations, with farming forming a central pillar of its economy. A vast number of families depend on cultivation for their livelihood, so when crops are affected by disease, the impact extends beyond individual farms and influences entire communities. Even so, many farmers in rural regions still rely on traditional methods to identify plant diseases, mainly by observing leaves and making judgments based on past experience. This method is time-consuming, varies in accuracy, and often depends on the availability of knowledgeable individuals nearby.

Recent technological progress has begun to address this challenge. In particular, deep learning techniques—systems capable of learning patterns from images—have shown strong performance in detecting disease symptoms in leaf photos, sometimes exceeding the accuracy of trained specialists. A widely used approach in this area is the Convolutional Neural Network (CNN), which processes images in small sections to recognize patterns such as shapes and color variations. However, CNNs can struggle to capture features distributed across larger portions of an image. An emerging alternative, the Vision Transformer (ViT), overcomes this limitation by evaluating the image as a whole and comparing relationships between different regions more effectively.

In this work, we integrated an ESP32-CAM, a compact Wi-Fi-enabled camera module, with a ViT model trained on a large dataset of leaf images. When a farmer positions the device near a plant and captures an image, it is transmitted online to a server where the model evaluates it in seconds, and the result is displayed on a user-friendly Gradio interface. The overall hardware expense was approximately ₹1,850, making the solution affordable and practical for small-scale farmers to adopt.

II. LITERATURE SURVEY

Ouamane et al., "Optimized Vision Transformers for Superior Plant Disease Detection," IEEE Access, 2025.
In this work, the authors explored multiple configurations of Vision Transformers, adjusting factors such as patch size, network depth, and attention mechanisms. Their findings showed that careful tuning of these parameters significantly improves detection accuracy, with the optimized ViT models consistently outperforming CNN-based approaches in most evaluations conducted.

Demilie K., "Plant Disease Detection and Classification Techniques," Springer, 2024.
This study compared various detection techniques and highlighted that although deep learning models perform

strongly under controlled laboratory conditions, their effectiveness often drops in real-world environments. Variations in lighting, camera positioning, and limited availability of diverse training data were identified as key challenges, leading the author to emphasize the importance of testing models in practical field scenarios.

Mamun et al., "Plant Disease Detection Using Self-Supervised Learning," IEEE Access, 2024.
The authors examined self-supervised learning approaches, where models learn meaningful representations without requiring extensive labelled datasets. While this method shows potential in addressing data scarcity, the study noted that such models demand high computational resources and may not generalize well when applied to different types of crops or environments.

Singh et al., "A Review of Imaging Techniques for Plant Disease Detection," Elsevier, 2020.
This review analyzed a range of imaging methods, including RGB imaging, thermal sensing, and hyperspectral techniques. The conclusion drawn was that standard visible-light cameras, when combined with robust deep learning models, provide a practical and cost-efficient solution, especially for farmers in developing regions.

Sharma and Kumar, "Rice Disease Identification using Pattern Recognition," ICCIT, 2018.
This earlier research demonstrated that isolating diseased regions of leaves and using those features within a neural network could achieve reasonable classification accuracy for rice diseases. It highlighted that even with simpler models and limited hardware, effective results are possible when the processing pipeline is well designed.

Summary:
Across these studies, a consistent trend can be observed: while model accuracy has improved significantly in controlled environments, translating that performance into affordable, real-world applications remains a challenge. None of the reviewed works integrated a low-cost microcontroller-based image capture system with a cloud-based Vision Transformer model in the manner proposed here, which is the specific gap this project aims to address.

III. PROBLEM STATEMENT AND OBJECTIVES

Problem Statement:
Farmers in rural India require a simple and affordable way to detect plant diseases without depending on expert assistance. Many existing solutions rely on costly equipment, uninterrupted internet access, or a certain level of understanding of machine learning, which makes them difficult to use in everyday farming conditions. Our objective was to create a device that is easy to operate, accessible to any farmer, and can be built for under ₹2,000.

Key gaps identified in this work:

- Advanced models often require high-performance GPUs, making them unsuitable for low-cost microcontroller-based systems.
- Available datasets do not adequately represent the diversity of diseases affecting crops in India.
- Only a limited number of solutions are designed to be portable and battery-operated for direct use in field conditions.

IV. OBJECTIVES

We defined the following clear objectives for this project:

1. Develop an image capture setup using the ESP32-CAM that can consistently take clear leaf images in outdoor environments.
2. Train a machine learning model capable of classifying leaves as healthy or identifying at least five different disease types.
3. Integrate the capture device with a cloud server so that image processing and prediction occur automatically after each capture.
4. Ensure the total hardware cost remains below ₹2,000 to keep the solution accessible to small-scale farmers.
5. Present the output on a straightforward web interface that can be accessed on a basic smartphone without requiring any application installation.

Academic Relevance: This project integrates microcontroller programming, wireless communication, image processing, and machine learning—key areas in both ECE and Computer Science studies. Developing and evaluating the system provided practical experience that extends far beyond theoretical knowledge gained from textbooks alone.

V. PROPOSED METHODOLOGY

The system operates as a sequence of four stages. First, the ESP32-CAM powers on, connects to a nearby Wi-Fi network, and remains ready for a capture signal. Once triggered, it captures a JPEG image of the leaf and promptly transmits it to the server using HTTP.

On the server side—hosted on Google Colab during development—a Python script receives the image and performs three preprocessing steps: resizing it to the required input dimensions, normalizing pixel values to a consistent scale, and applying a light noise reduction filter to account for dust or moisture on the lens. The processed image is then passed to the Vision Transformer model, which generates a ranked set of possible disease classes along with confidence percentages for each prediction.

The highest-ranked result is converted into a concise text output and sent to the Gradio interface, where it is displayed within a few seconds of image capture. If the confidence level falls below a predefined threshold, the system marks the prediction as uncertain instead of showing a potentially incorrect result.

Working Flow

1. Download and organise the plant disease image dataset from Kaggle.
2. Train the Vision Transformer (ViT) model on this dataset using Google Colab.
3. Power on the ESP32-CAM, allowing it to connect to Wi-Fi automatically.
4. Capture an image of a leaf and send it to the inference server.
5. The server preprocesses the image by resizing, normalising, and reducing noise.
6. The model analyses the image and produces a classification along with a confidence score.
7. The result and its confidence level are displayed on the Gradio web interface.

VI. COMPONENT REQUIREMENTS

Hardware:

ESP32-CAM:

This compact module combines a dual-core processor, 4 MB RAM, an integrated 2-megapixel camera, and Wi-Fi connectivity in a board about the size of a matchbox. Operating at 240 MHz, it is capable of handling both image capture and wireless communication at the same time. Priced at roughly ₹600, it offers excellent performance for its cost.

LM2596 Voltage Regulator:

The ESP32-CAM requires a stable 5 V power supply. Since battery voltage tends to drop during use, a direct connection can lead to unstable operation or unexpected resets. The LM2596 switching regulator accepts an input range of approximately 4 V to 40 V and reliably converts it to a constant 5 V output, ensuring consistent performance even as battery levels decrease.

18650 Lithium-Ion Batteries:

Two 18650 lithium-ion cells were connected in series, providing a nominal voltage of 7.4 V with a capacity of 2,200 mAh. This setup powers the regulator and allows the system to operate for several hours in field conditions with intermittent usage before requiring a recharge.

Software:

Google Colab was used to access free GPU resources for both training the model and performing inference. Python, together with PyTorch and OpenCV, was used to implement the machine learning and image processing components. The training data was obtained from Kaggle's Plant Village dataset. Gradio made it possible to convert the model into a functional web application with minimal code.

VII. CIRCUIT EXPLANATION

The wiring setup is intentionally kept straightforward. Two 18650 batteries are placed in a holder and connected in series, where the positive terminal of one cell connects to the negative terminal of the other. This produces a combined output of about 7.4 V when fully charged, which is supplied to the input of the LM2596 regulator module. The onboard trimmer potentiometer is adjusted until the output is stabilized at exactly 5.0 V, verified using a multimeter.

From the regulator's output, two wires are connected to the ESP32-CAM: the positive (red) wire is linked to the 5 V pin, and the negative (black) wire is connected to GND. The camera module is attached via its ribbon cable into the designated connector on the board. For initial programming, an FTDI adapter is connected to the serial pins to upload the firmware.

Once programming is complete, the FTDI module is removed, and the system operates independently, powering up and running automatically whenever it is supplied with power.

VIII. RESULTS

We evaluated the assembled system under different conditions, including indoor artificial lighting, bright outdoor sunlight, and partial shade. In well-lit environments, the ESP32-CAM captured clear and properly exposed images, allowing the model to make predictions with high confidence. The Wi-Fi connection remained reliable within the typical range of a standard router, and images were transmitted to the server in about half a second.

The model accurately classified the leaf condition in six out of eight test cases. The two incorrect predictions occurred when image quality was poor: one image was overexposed due to strong sunlight, reducing visible color differences, while the other involved a leaf where Early Blight and Late Blight appeared very similar even to the human eye. When these samples were captured again under improved conditions, the model produced correct results.

The power system performed consistently during testing. The voltage supplied to the ESP32-CAM remained stable within 0.05 V of the required 5 V, and no unexpected resets were observed, even during prolonged operation with image captures taken every thirty seconds.

OBSERVATION TABLE

Table 1 presents each test image along with its actual disease label, the model's predicted result, the corresponding confidence score, and an indication of whether the prediction was correct.

Table 1: Disease Prediction Results

Sl. No	Input Image	Actual Condition	Predicted Output	Confidence (%)	Result
1	Leaf Image 1	Healthy	Healthy	96%	Correct
2	Leaf Image 2	Early Blight	Early Blight	91%	Correct
3	Leaf Image 3	Late Blight	Late Blight	93%	Correct
4	Leaf Image 4	Healthy	Leaf Spot	74%	Incorrect
5	Leaf Image 5	Leaf Spot	Leaf Spot	88%	Correct
6	Leaf Image 6	Bacterial Spot	Bacterial Spot	90%	Correct
7	Leaf Image 7	Early Blight	Late Blight	78%	Incorrect
8	Leaf Image 8	Healthy	Healthy	95%	Correct

X. OBJECTIVES ACCOMPLISHED

Considering the results overall, the system achieved the objectives set at the beginning. The ESP32-CAM was able to capture clear images under various outdoor conditions, the Wi-Fi communication remained dependable, and the model performed well in identifying common disease categories with satisfactory accuracy. The two incorrect predictions highlight an important limitation—image quality plays a critical role—but they also indicate where future improvements should be focused.

From a hardware perspective, the combination of batteries and voltage regulation worked reliably. The device operated continuously for more than four hours on a single charge without any signs of instability, which is sufficient for typical field use. The Gradio interface was also user-friendly, allowing non-technical users to quickly understand both the prediction and its confidence score without requiring additional guidance.

XI. ACTION PLAN

Table 2: Project Action Plan

Sl. No	Task	Description	Duration	Status
1	Requirement Analysis	Understanding project goals and system needs	1 Week	Completed
2	Literature Review	Reading research papers on plant disease detection	2 Weeks	Completed
3	Dataset Collection	Fetching and organizing the Kaggle plant disease dataset	1 Week	Completed
4	Model Training	Training the ViT model on Google Colab	2 Weeks	Completed

5	Hardware Setup	Assembling ESP32-CAM, battery and regulator	1 Week	Completed
6	Prototype Development	Connecting hardware with software end-to-end	3 Weeks	Completed
7	Testing & Validation	Running real-time tests and measuring accuracy	2 Weeks	Completed
8	Result Analysis	Studying outputs and refining the model	1 Week	Completed
9	Documentation	Writing the report and preparing diagrams	2 Weeks	Completed
10	Final Review & Submission	Final check and project submission	1 Week	Completed

XII. COST OF THE PROJECT

Table 3: Bill of Materials

Sl. No	Component	Description	Cost (INR)
1	ESP32-CAM	Camera + Wi-Fi Module	₹600
2	18650 Li-Ion Battery ×2	Primary Power Source	₹300
3	Battery Holder	Mounts the Batteries	₹100
4	LM2596 Voltage Regulator	Step-Down DC Converter	₹100
5	Jumper Wires	Circuit Connections	₹50
6	FTDI Programmer	Code Flashing Interface	₹500
7	Miscellaneous	Casing, Connectors	₹200
	Total		**₹1,850**

Future Scope

The current system serves as a functional prototype, and there are several meaningful ways to enhance it further. One of the most significant improvements would be to run the model directly on the ESP32 instead of relying on cloud-based processing. By applying model quantization—reducing the precision of its parameters—the model size can be minimized enough to fit on the device, enabling offline usage in areas without reliable internet connectivity.

Another important extension would be to broaden the range of crops and diseases the model can identify. At present, it is primarily trained on tomato leaves; expanding it to include crops such as wheat, rice, and cotton would increase its usefulness across diverse agricultural settings. Incorporating additional sensors, such as a soil moisture sensor and a temperature probe, could also allow the system to monitor environmental conditions along with disease detection, providing more comprehensive insights into crop health.

From a software perspective, converting the Gradio interface into a dedicated mobile application with offline support and voice assistance in regional languages would improve accessibility, especially for users who may find it difficult to read English text on a screen.

REFERENCES

1. Ouamane A, Chouchane A, Himeur Y, Miniaoui S, Atalla S, Mansoor W, Al-Ahmad H, "Optimized Vision Transformers for Superior Plant Disease Detection," IEEE Access, Vol. 13, pp. 48552–48569, 2025.
2. Demilie K, "Plant Disease Detection and Classification Techniques: A Comparative Study of the Performances," Springer, Vol. 11, pp. 1–43, 2024.
3. Mamun A et al., "Plant Disease Detection Using Self-Supervised Learning: A Systematic Review," IEEE Access, Vol. 12, pp. 171931–171950, 2024.
4. Singh V, Sharma N, Singh S, "A Review of Imaging Techniques for Plant Disease Detection," Artificial Intelligence in Agriculture, Vol. 4, pp. 229–242, 2020.
5. Sharma R, Kumar S, "Rice Disease Identification Using Pattern Recognition Techniques," Proc. ICCIT 2018, Vol. 2, pp. 1–6, 2018.