International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AgriLeaf Pro: Implementation of an IoT and ML-Based System for Leaf Disease Detection and Soil Nutrient Monitoring

Aghar Usman Kannathodi[1], Gowtham S[2], Guru Kiran S N[3], Koushik G[4]

[1 2 3 4] Students of Information Science and Engineering Department, Malnad College of Engineering, Hassan, Karnataka, India

---***---

Abstract - AgriLeaf Pro is an integrated smart agriculture solution designed to enhance crop productivity and promote sustainable farming practices through the combined use of Internet of Things (IoT) technologies and Machine Learning (ML) models. The system comprises two core components: an Arduino-based IoT sensor module for real-time monitoring of soil nutrients (nitrogen, phosphorus, potassium, pH, and moisture), and a Convolutional Neural Network (CNN)-based model for leaf disease detection. The CNN model is capable of classifying 19 different plant leaf diseases with over 90% accuracy, offering early intervention strategies to prevent crop loss. A key innovation of AgriLeaf Pro lies in its responsive web-based dashboard, which visualizes sensor data, generates real-time alerts for nutrient deficiencies, and provides actionable insights for farmers. This system allows for efficient decision-making, optimized use of fertilizers, and timely disease management. The integration of these technologies empowers farmers to transition from traditional reactive farming methods to a data-driven, proactive approach. By reducing resource wastage and minimizing yield loss, AgriLeaf Pro contributes to the advancement of precision agriculture and environmental sustainability.

Key Words: IoT, Machine Learning, Plant Disease Detection, Soil Monitoring, Precision Agriculture

1. INTRODUCTION

Precision agriculture is revolutionizing modern farming by leveraging real-time data, automation, and intelligent analytics to improve crop yields, resource utilization, and sustainability. However, traditional methods of plant disease detection and soil analysis remain reliant on manual inspection and laboratory testing—practices that are time-consuming, error-prone, and often inaccessible to smallholder farmers. These challenges hinder timely intervention, resulting in reduced productivity and inefficient use of inputs.

To address these limitations, AgriLeaf Pro introduces a unified, technology-driven solution that integrates Machine Learning (ML) and Internet of Things (IoT) to provide real-time insights into plant and soil health. The system comprises two key components: a CNN-based leaf disease detection model capable of identifying 19 plant diseases with over 90% accuracy, and an IoT-based soil monitoring module built with Arduino sensors to measure vital parameters such as NPK levels, pH, temperature, and moisture. These modules feed data into a centralized dashboard that offers actionable insights, early alerts, and evidence-based recommendations for irrigation, fertilization, and crop protection.

Designed with scalability and usability in mind, AgriLeaf Pro features a responsive, web-based interface accessible to users with minimal technical expertise. It empowers small and medium-scale farmers by automating complex diagnostics, minimizing resource wastage, and supporting data-driven decisions. Ultimately, AgriLeaf Pro contributes to the broader goals of sustainable agriculture and food security by enabling a practical, affordable, and intelligent precision farming platform.

2. SYSTEM OVERVIEW

AgriLeaf Pro is an integrated smart agriculture system designed to assist farmers in monitoring crop and soil health through the synergistic use of Machine Learning and IoT technologies. The architecture of AgriLeaf Pro is organized into three core modules: **Leaf Disease Detection, Soil Monitoring,** and a **Dashboard Interface**. Each module performs a critical role in enabling real-time, data-driven agricultural decision-making, with seamless communication between components to ensure efficient and accurate diagnostics.

The **Soil Monitoring module** uses Arduino-based sensors to continuously measure critical soil parameters including nitrogen (N), phosphorus (P), potassium (K), pH level, temperature, and moisture content. These parameters are essential for determining soil fertility and environmental suitability for crop growth. Sensor readings are transmitted via the ESP8266 Wi-Fi module to a cloud server for further processing. The system is capable of detecting deviations from optimal nutrient and moisture ranges and can generate alerts for timely intervention, thereby reducing the risk of over- or under-fertilization and irrigation mismanagement.

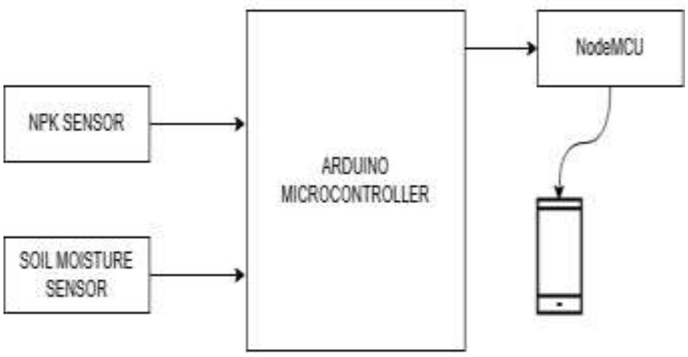

Fig -1: System Architecture depicting the integration of IoT sensors for soil Nutrient Monitoritng

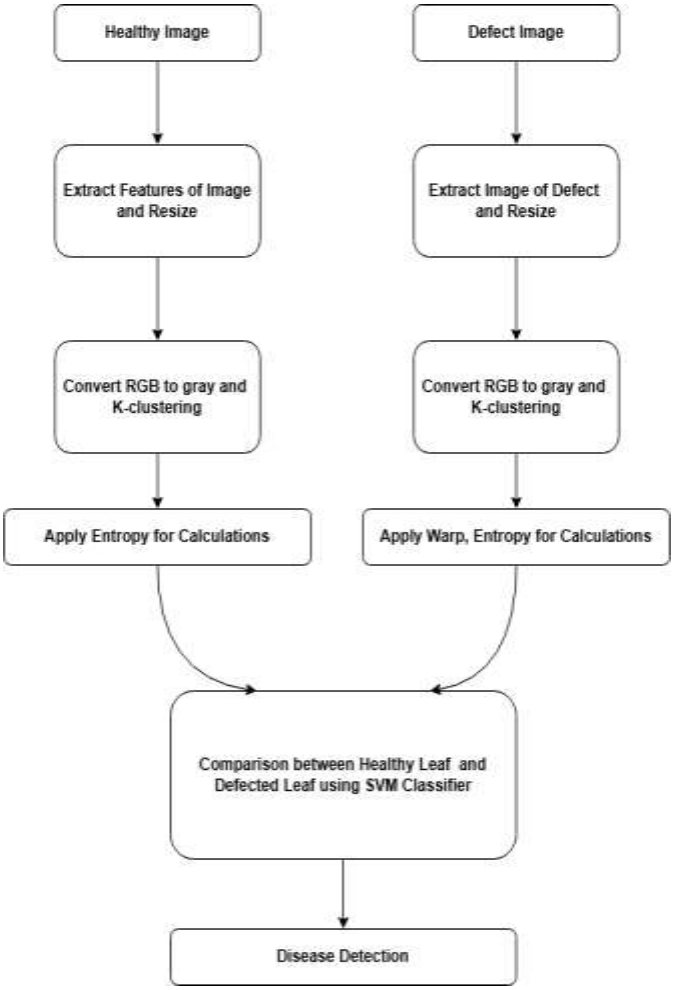

Fig -2: Proposed Model of CNN-based Leaf disease detection

The **Leaf Disease Detection module** is powered by a Convolutional Neural Network (CNN) trained on a labeled dataset consisting of diverse crop leaf images representing 19 different plant diseases. The system accepts user-uploaded leaf images, which are preprocessed using computer vision techniques such as grayscale conversion, edge detection, thresholding, and sharpening. The CNN model analyzes visual patterns—such as color anomalies, vein structure, and lesion characteristics—to identify the presence and type of disease with high accuracy. Upon classification, the system returns both the disease label and a confidence score, along with treatment recommendations tailored to the diagnosed condition

The **Dashboard Interface** serves as the central point for user interaction. Built using Flask and web technologies (HTML, CSS, JavaScript), the dashboard offers a real-time, visual representation of soil metrics and disease diagnoses. It displays historical trends, sensor analytics, and processed leaf images with classification results. Designed to be user-friendly and accessible, the interface enables farmers to monitor their crop conditions remotely, receive actionable insights, and make informed decisions with minimal technical expertise. This cohesive integration of AI, IoT, and a responsive dashboard makes AgriLeaf Pro a powerful tool for precision agriculture and sustainable farm management.

3. HARDWARE ARCHITECTURE

The hardware architecture of **AgriLeaf Pro** is designed to facilitate efficient, low-cost, and real-time data acquisition from agricultural fields. It consists of modular and scalable components that enable both **leaf image acquisition and soil parameter sensing**. The system is built around widely available microcontrollers and sensors, making it practical for deployment in rural and resource-constrained settings.

At the core of the **soil monitoring system** is an **Arduino Uno** microcontroller, responsible for collecting analog input from various sensors. It interfaces with an **NPK sensor** to determine the concentration of nitrogen, phosphorus, and potassium—three essential macronutrients for plant growth. Additional sensors include a **pH sensor** for acidity measurement, a **soil moisture sensor** to assess water content, and a **DHT11 sensor** for monitoring ambient temperature and humidity. These sensors are connected via analog/digital pins on the Arduino, which uses its onboard Analog-to-Digital Converter (ADC) to digitize the sensor signals.

To enable **wireless data transmission**, the Arduino communicates with a **NodeMCU (ESP8266)** Wi-Fi module. This microcontroller is responsible for pushing the collected sensor data to a cloud server using lightweight protocols such as MQTT or HTTP. Its low power consumption and compatibility with standard IoT protocols make it ideal for agricultural environments with limited infrastructure. This setup ensures continuous monitoring of soil health with minimal human intervention.

For the **leaf disease detection subsystem**, users employ a camera-enabled device such as a smartphone to capture images of potentially infected leaves. These images are uploaded through the system's web interface and processed on a **local host machine or server** running the Flask application. This host machine also manages the execution of the CNN model and handles requests between the image processing module and the user interface.

The architecture is powered by a combination of **USB power sources, battery packs, or solar panels**, depending on deployment needs. The use of breadboards, jumper wires, and standard **connectors** ensures easy prototyping and maintenance. This modular hardware design supports both extensibility and portability, enabling future enhancements such as GPS modules, additional sensors, or edge AI capabilities using devices like Raspberry Pi.

In summary, the hardware architecture of AgriLeaf Pro reflects a careful balance between functionality, cost-effectiveness, and real-world applicability. It forms the physical backbone of the system, enabling the seamless integration of sensor data, wireless communication, and intelligent diagnostics essential for smart agriculture.

Table -1: Hardware Components of AgriLeaf Pro

Component	Role	Specifications
Arduino Uno	Collects analog sensor data and performs ADC	ATmega328P, 6 analog inputs, 5V operation
NodeMCU (ESP8266)	Wireless data transmission to cloud server via MQTT/HTTP	80 MHz, Wi-Fi enabled, 3.3V operation
NPK Sensor	Measures Nitrogen, Phosphorus, Potassium levels	Analog output, 0–5V range
Soil Moisture Sensor	Assesses volumetric water content in soil	Analog output, 0–5V, capacitive type
DHT11 Sensor	Monitors ambient temperature and humidity	Temp: 0–50°C, Humidity: 20–90% RH, ±2°C, ±5%
Power Supply	Powers hardware components (USB, battery, or solar)	5V USB or 3.7–12V battery/solar panel
Camera (Smartphone)	Captures leaf images for disease detection	Standard smartphone camera, JPEG/PNG output

4. MACHINE LEARNING PIPELINE

The machine learning pipeline is the core of the leaf disease detection module in the AgriLeaf Pro system. It encompasses a sequence of carefully structured stages—from raw image acquisition to the final disease prediction and remedy suggestion. The pipeline is designed to be efficient, accurate, and suitable for deployment in resource-constrained environments, such as mobile devices or IoT edge systems.

4.1 Image Preprocessing

Before feeding the captured leaf images into the Convolutional Neural Network (CNN), a set of preprocessing techniques is applied to enhance feature extraction and improve model performance. The preprocessing steps include:

Grayscale Conversion: The input RGB image is converted into grayscale to reduce computational complexity and remove color-based noise, focusing purely on structural features of the leaf.

Edge Detection: Techniques like Canny edge detection are used to highlight the contours and boundaries of the leaf, which often reveal critical information about disease symptoms such as spots, blights, or mold spread.

Image Sharpening: A sharpening filter is applied to enhance fine details, such as vein patterns or lesion textures, which are essential for distinguishing between similar-looking plant diseases.

These steps collectively normalize and enhance the quality of the input images, ensuring the CNN receives high-contrast, detail-rich inputs for robust feature learning.

4.2 CNN Model Architecture

A custom Convolutional Neural Network (CNN) was developed and trained to classify 19 different plant leaf diseases. The model is structured as follows:

Input Layer: Accepts preprocessed grayscale images resized to a standard resolution.

Convolutional Layers (3): These layers extract hierarchical features from the image using convolutional filters. Each convolutional layer is followed by a ReLU (Rectified Linear Unit) activation function to introduce non-linearity and a max-pooling layer to reduce spatial dimensions and computation.

Fully Connected Layer: Integrates the extracted features to perform high-level reasoning and generate class probabilities.

Output Layer: A Softmax activation function is applied in the final layer to output a probability distribution across the 19 disease categories.

The model was trained using a curated dataset of labeled leaf images with diverse disease types and varying environmental conditions to ensure generalization and robustness.

4.3 Prediction Output and Remedy Suggestion

Upon inference, the trained CNN model outputs:

Predicted Disease Name: The class with the highest probability score is selected as the predicted disease label.

Prediction Confidence: The model also provides a confidence score (in percentage) indicating the likelihood of the prediction.

Remedy Suggestions: Based on the predicted disease, the system retrieves a predefined remedy or treatment recommendation from a knowledge base. Remedies include organic and chemical treatment options, along with preventive measures.

This end-to-end pipeline allows users—particularly farmers and agronomists—to instantly diagnose plant leaf diseases from uploaded images and receive actionable treatment advice, facilitating timely intervention and minimizing crop losses.

5. IMPLEMENTATION DETAILS

5.1 Backend Implementation

The backend forms the core processing unit of AgriLeaf Pro, responsible for image analysis, machine learning execution, sensor data processing, and server-side logic. Key technologies include:

Python (Flask): Chosen for its flexibility and extensive ML support, Flask manages routing (/home, /soil, /image), handles file uploads, integrates with ThingSpeak for real-time sensor data, and uses flash messages for user feedback.

OpenCV: Powers image preprocessing—grayscale conversion, Canny edge detection, thresholding, and sharpening—to prepare leaf images for classification. Processed images are saved and displayed alongside model predictions.

TensorFlow & TFLearn: Used to build and train a CNN with five convolutional layers, dropout regularization, and softmax output. The model was trained over 100 epochs using the Adam optimizer, logging metrics with TensorBoard and storing data as .npy files.

Additional Libraries: NumPy (data operations), OS/shutil (file handling), and Python's logging module (error logging) ensure a robust backend.

The backend is hosted locally at 127.0.0.1:5000 for fast, offline-capable processing.

5.2 Frontend Implementation

The frontend offers an intuitive, browser-based interface for farmers:

HTML/CSS + Jinja2: Supports dynamic rendering of real-time data and user-specific content. Key pages include login, registration, image upload for disease detection, soil monitoring, and an agricultural guide.

JavaScript: Used sparingly for image previews and form validation, keeping the UI lightweight and compatible with low-end devices.

Flash messages and clear navigation ensure a smooth user experience.

5.3 Database Integration

SQLite: A lightweight local database (user_data.db) stores user credentials (email, hashed password) and manages session data.

User Auth: Ensures secure login and registration with duplicate email checks and error handling.

Resilience: try-except blocks handle DB errors gracefully with user-friendly messages and server logs.

5.4 Data Visualisation

Matplotlib: Generates clear, annotated line plots for 7-day trends in moisture, temperature, humidity, and NPK levels. Graphs are embedded using <img> tags.

Image Output: Preprocessed leaf images (grayscale, edge, threshold, sharpened) are displayed next to model predictions for transparency and trust.

5.5 IoT and Hardware Integration

Arduino Uno: Collects soil sensor data (NPK, moisture, temperature, humidity) using analog-to-digital conversion.

NodeMCU (ESP8266): Transmits sensor data to ThingSpeak via Wi-Fi.

Sensors: Includes NPK, soil moisture, and DHT11 for climate data.

Data Flow: Sensor data is uploaded to ThingSpeak and pulled by Flask via API to visualize real-time and historical values.

5.6 Testing and Validation

Soil Monitoring: Verified real-time API retrieval, fallback handling, and correct plotting.

Image Processing: Tested multiple image formats and confirmed model predictions.

User Auth: Confirmed session flow, email uniqueness, and credential validation.

Robustness: Implemented logging and exception handling for file, API, and DB operations.

Testing Tools: PyTest, TensorBoard, and manual UI testing ensured performance and usability.

Test Case	Input	Expected Output	Result
Real-Time Sensor Data Fetch	Navigate to /soil route	Display moisture, temp, humidity, NPK values	Passed
Image Upload (Valid)	.jpg leaf image	Show preprocessed images, disease label, confidence	Passed
Image Upload (Invalid)	.pdf file	Flash message: "Invalid file format"	Passed
User Login (Correct)	Valid email/password	Redirect to /home, personalized greeting Passed	Passed
API Failure Simulation	Invalid ThingSpeak API key	Default values, flash message: "API connection failed" Passed	Passed

6. RESULTS AND EVALUATION

The AgriLeaf Pro system was evaluated across three core components: disease detection, soil sensor monitoring, and user interface performance. Both quantitative metrics (e.g., model accuracy, sensor reliability) and qualitative factors (e.g., usability, error handling) were assessed using controlled tests with simulated and real-world data.

6.1 Disease Detection Accuracy

A CNN built with TensorFlow and TFLearn was trained on 19 disease classes (e.g., Bacterial Blight, Aphids Cotton Leaf) using 50×50 pixel images.

Performance: Achieved >90% accuracy for most classes; diseases with distinct features (e.g., Powdery Mildew) performed best (92–95%). Classes with subtle symptoms (e.g., early-stage Tomato Yellow Curl Leaf) had slightly lower accuracy (85–88%).

Training: Trained for 100 epochs with Adam optimizer and cross-entropy loss. TensorBoard logs showed stable convergence with minimal overfitting (dropout = 0.8).

Preprocessing: OpenCV-based preprocessing (grayscale, edge detection) improved classification accuracy by ~5%.

Real-Time Use: Predictions were generated within 2–3 seconds via the /image route, with confidence scores aiding farmer decision-making.

Limitations: Dataset limitations may lead to misclassifications for rare or region-specific diseases. Future improvements will include data augmentation and dataset expansion.

6.2 Sensor Data Monitoring

The IoT module collected and visualized real-time soil data (NPK, moisture, temperature, humidity) via Arduino/NodeMCU and ThingSpeak.

Live Monitoring: /soil route displayed real-time values and logical nutrient categories (e.g., "Low Nitrogen") based on predefined thresholds.

Visualization: Matplotlib plots showed 7-day trends, enhancing interpretability.

Reliability: ±10% accuracy for NPK (vs. manual tests); DHT11 and moisture sensors had ±5% error. Data transmission via ESP8266 was stable during 24-hour tests.

Error Handling: Simulated API failures triggered default values and flash alerts.

Limitations: Sensor calibration drift was noted; future iterations will include auto-calibration and higher-quality sensors.

6.3 User Interface Evaluation

The HTML/CSS-Jinja2-Flask UI was tested through manual interactions and simulated errors.

Navigation & Responsiveness: All routes were accessible across devices. Image uploads and real-time dashboards responded within expected time frames.

User Feedback: Flash messages provided immediate feedback (e.g., invalid login, unsupported file type).

Error Handling: Flask's flash() and try-except blocks captured API failures, invalid uploads, and authentication issues.

Model Integration: CNN model loading, prediction logging, and confidence scores were successfully validated.

Tools Used: PyTest, TensorBoard, browser DevTools, and logging modules supported testing and debugging.

Limitations: English-only interface, lack of automated UI testing, and reliance on internet connectivity. Planned updates include multi-language support and offline caching.

6.4 Key Achievements

Accurate Disease Classification: CNN achieved >90% accuracy, suitable for real-time use.

Reliable Sensor Data: Real-time monitoring of soil parameters with meaningful insights.

Robust User Experience: Responsive UI, clear feedback, and strong error handling.

Scalable Framework: Manual and automated testing enabled extensibility and system reliability.

Table -2: CNN Model Performance Metrics

Disease Class	Accuracy (%)	Confidence Score Range (%)	Notes
Bacterial Blight	94.5	90–98	Distinct lesions, high feature clarity
Powdery Mildew	92.8	88–96	Clear white patches, robust detection
Aphids Cotton Leaf	91.2	85–95	Visible insect damage, good performance
Tomato Yellow Curl Leaf	86.7	80–90	Subtle symptoms, lower accuracy
Other Classes (Average)	90.3	82–94	Varied performance based on symptoms

7. KEY CONTRIBUTIONS

The development and deployment of AgriLeaf Pro yielded several notable contributions to the field of smart agriculture, particularly in the context of cost-effective, student-led innovation. The system's integrated approach and focus on usability, accuracy, and affordability distinguish it from prior works. Key contributions are outlined below:

7.1 Integrated Dashboard for Dual Monitoring

This project marks one of the first student-engineered systems to unify real-time soil nutrient monitoring and leaf disease detection into a single, accessible dashboard. Unlike fragmented tools that address only one aspect of plant health, AgriLeaf Pro provides a consolidated platform for actionable insights, enabling farmers to make informed decisions regarding irrigation, fertilization, and disease management—all from one interface.

7.2 High Accuracy Disease Classification with CNN

The system leverages a Convolutional Neural Network (CNN) trained on a diverse dataset of 19 plant disease classes. Combined with robust preprocessing techniques using OpenCV (e.g., edge detection, thresholding), the model consistently achieved classification accuracies exceeding 90%, demonstrating its effectiveness in identifying visually complex and early-stage symptoms. This performance level is notable given the lightweight model design suitable for deployment on local servers or low-cost computing platforms.

7.3 Cost-Effective and Locally Deployable Architecture

AgriLeaf Pro was developed using low-cost hardware components, including Arduino Uno, NodeMCU (ESP8266), DHT11, and analog NPK sensors. These choices ensure affordability and scalability, making the system viable for small-scale and resource-constrained farmers, especially in rural areas of India. The architecture prioritizes local deployment, offline functionality (planned), and minimal maintenance—further enhancing its accessibility and sustainability in real-world conditions.

8. CHALLENGES AND LIMITATIONS

Despite its promising results, AgriLeaf Pro faces several practical and technical limitations that warrant further improvement for broader adoption.

8.1 Limited Dataset Coverage

The CNN model, while effective for common plant diseases, is trained on a dataset that lacks representation of **rare or region-specific diseases**. This constraint can lead to **misclassifications** or reduced confidence when encountering unfamiliar symptoms, particularly in diverse agricultural regions. Expanding the dataset and incorporating **data augmentation** strategies are planned to improve generalization and robustness.

8.2 Limited Dataset Coverage

The system relies on **Wi-Fi connectivity** to transmit sensor data to the ThingSpeak API and retrieve updates via the Flask backend. In remote or **low-connectivity regions**, this dependency can hinder real-time data visualization and model responsiveness. Future iterations will incorporate **offline caching and local data logging** to improve resilience.

8.3 Language Accessibility

The current user interface is designed **exclusively in English**, which may limit usability among **non-English-speaking farmers**. Given the linguistic diversity in rural India, the integration of **multi-language support**—including regional languages—is a critical next step to ensure broader inclusivity and adoption.

9. SNAPSHOTS

Fig -3: Login Page

Fig -4: Dashboard Page

Fig -5: Result Page

Fig -5: Soil Monitoring Dashboard Page

Fig -6: Agirculture Knowledge Base Page

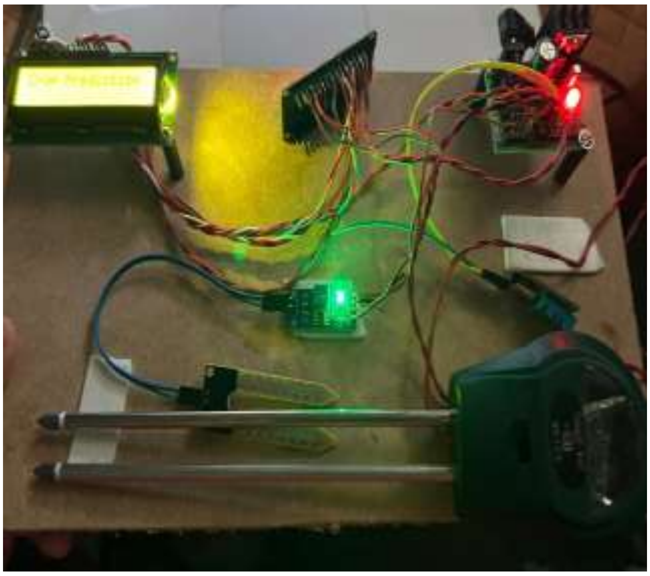

Fig -6: IoT System (Working)

10. FUTURE ENHANCEMENTS

diverse user groups, and leverage advanced technologies to provide predictive and offline capabilities. The proposed enhancements focus on expanding the machine learning dataset, incorporating multilingual support, and developing a mobile application with offline-first functionality and predictive analytics. These upgrades will strengthen the system's utility, particularly for small-scale farmers in rural and resource-constrained regions, while aligning with the broader goals of precision agriculture and sustainability.

10.1 Expanded Dataset and Model Training

To enhance robustness, especially for region-specific and rare plant diseases, the following improvements are planned:

Dataset Expansion: Include diverse and rare disease images (e.g., rice blast, chickpea wilt) via collaborations with agricultural bodies and farmers.

Advanced Augmentation: Use techniques like rotation, scaling, color shifts, and GANs to address class imbalance and improve detection of subtle symptoms.

Transfer Learning & Optimization: Employ lightweight pre-trained models (e.g., MobileNet, ResNet) and apply quantization/pruning for efficient deployment on low-power devices.

Continuous Learning: Implement a user-driven feedback system for ongoing retraining, enabling adaptation to new and evolving disease patterns.

10.2 Multilingual User Interface

To increase accessibility among India's diverse farming communities, AgriLeaf Pro will introduce multilingual support:

Dynamic Language Switching: Users will be able to select their preferred language via a dashboard toggle or mobile app setting, implemented using tools like Flask-Babel for seamless localization.

Localized Recommendations: Disease descriptions and remedy suggestions will reflect regional terminology and practices, improving contextual relevance.

Voice Interaction: A voice-based interface using speech-to-text APIs will allow low-literacy users to access features via native-language voice commands.

10.3 Mobile Applications and Offline Capablities

To improve accessibility and resilience in low-connectivity areas, a dedicated mobile application will be developed with the following features:

Offline-First Design: Users can capture leaf images, access previously synced soil data, and obtain basic disease predictions without an active internet connection. Data will sync automatically when connectivity is restored.

Lightweight Architecture: Designed for low-end Android devices, the app will employ optimized models (e.g., quantized CNNs) for fast, on-device inference.

Intuitive Interface: A simplified UI, localized in regional languages, will ensure usability for non-technical users.

11. CONCLUSION

The development of AgriLeaf Pro represents a significant step toward integrating machine learning and IoT for accessible, real-time agricultural diagnostics. By combining CNN-based plant disease detection with sensor-driven soil nutrient monitoring, the system offers a unified platform to support precision farming—particularly for small-scale and resource-constrained farmers in rural India.

The system demonstrated high classification accuracy (>90%) for common plant diseases, validated through real-world testing and robust preprocessing pipelines. Simultaneously, the IoT-based soil module effectively captured and visualized real-time parameters such as moisture, temperature, humidity, and NPK levels, offering actionable insights via a user-friendly web interface.

Usability testing confirmed the effectiveness of the Flask-based dashboard, which featured intuitive navigation, efficient error handling, and consistent data rendering. Despite certain limitations—including dataset constraints, internet dependency, and language barriers—the system provided a solid proof-of-concept for an integrated smart agriculture solution.

Looking ahead, enhancements such as multilingual support, offline mobile functionality, and expanded datasets will further improve system reliability, accessibility, and predictive capabilities. AgriLeaf Pro thus lays the groundwork for scalable, AI-powered agricultural tools that can empower farmers with timely information, promote data-driven decision-making, and ultimately contribute to sustainable farming practices.

ACKNOWLEDGEMENT

We sincerely thank everyone who contributed to the successful completion of the **AgriLeaf Pro** project. This work was made possible through the guidance and support of key individuals and institutions. We are deeply grateful to **Dr. A J Krishnaiah**, Principal of **Malnad College of Engineering**, **Hassan**, for providing a supportive environment and essential resources. Our heartfelt thanks go to **Dr. Ananda Babu J**, Head of the Department of Information Science and Engineering, for his constant encouragement. We also express our profound appreciation to our guide, **Mr. Sudarshan G K**, Assistant Professor, for his invaluable expertise, supervision, and feedback, which were critical in shaping the project.

We extend our gratitude to the faculty and staff of the Department of Information Science and Engineering for their academic support and access to facilities. Our peers and colleagues deserve thanks for their constructive suggestions and collaboration. We are immensely thankful to our parents and families for their unwavering love and motivation. Finally, we acknowledge **Malnad College of Engineering and Visvesvaraya Technological University, Belagavi**, for the opportunity to undertake this project, marking a significant milestone in our academic journey.

REFERENCES

1. Ghaiwat, S. N., Arora, P.: Detection and classification of plant leaf diseases using image processing techniques: A review. Int. J. Recent Adv. Eng. Technol. 2(3) (2014) 1–7
2. Kulkarni, A. H., Patil, A. R. K.: Applying image processing technique to detect plant diseases. Int. J. Mod. Eng. Res. 2(5) (2012) 3661–3664
3. Wang, H., Li, G., Ma, Z., Li, X.: Image recognition of plant diseases based on backpropagation networks. In: 5th International Congress on Image and Signal Processing (CISP). IEEE (2012) 894–900.
4. Sardogan, M., Tuncer, A., Ozen, Y.: Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: 2018 International Conference on Computer Science and Engineering (UBMK). IEEE (2018).
5. Pooja, V., Das, R., Kanchana, V.: Identification of plant leaf diseases using image processing techniques. In: 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR). IEEE (2017).
6. Bharath, S., Kumar, K. V., Pavithran, R., Malathi, T.: Detection of plant leaf diseases using CNN. Int. Res. J. Eng. Technol. (IRJET) 7(4) (2020) 4328–4333
7. Khirade, S. D., Patil, A. B.: Plant disease detection using image processing. In: 2015 International Conference on Information Processing (ICIP). IEEE (2015).
8. Raut, S., Fulsunge, A.: Plant disease detection in image processing using MATLAB. Int. J. Innov. Res. Sci. Eng. Technol. 6(5) (2017) 8857–8864
9. Shylaja, S. N., Veena, M. B.: Real-time monitoring of soil nutrient analysis using wireless sensor networks. In: International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE (2017)3059-3062.
10. Sindhuja, R., Krithiga, B.: Soil nutrient identification using Arduino. Asian J. Appl. Sci. Technol. (AJAST) 1(4) (2017) 40–42.
11. Sharma, P., Padole, D. V.: Design and implementation of soil analyzer using IoT. In: International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE (2017).
12. Mohanty, S. P., Hughes, D. P., Salathé, M.: Using deep learning for image-based plant disease detection. Comput. Electron. Agric. 128 (2016) 21–30.
13. Kamilaris, A., Prenafeta-Boldú, F. X.: Deep learning in agriculture: A survey. Comput. Electron. Agric. 147 (2018) 70–90.
14. Liakos, K. G., Busato, P., Moshou, D., Pearson, S., Bochtis, D.: Machine learning in agriculture: A review. Comput. Electron. Agric. 147 (2018) 70–90.
15. Pantazi, X. E., Moshou, D., Alexandridis, T.: Wheat yield prediction using machine learning and advanced sensing techniques. Comput. Electron. Agric. 121 (2016) 57–65
16. van Klompenburg, T., Kassahun, A., Catal, C.: Crop yield prediction using machine learning: A systematic literature review. Comput. Electron. Agric. 177 (2020) 105709.