International Journal of Research in Engineering 2026; 8(1): 109-114

E-ISSN: 2664-8784
P-ISSN: 2664-8776
Impact Factor: RJIF 8.26
IJRE 2026; 8(1): 109-114

www.engineeringpaper.net
Received: 10-01-2026
Accepted: 15-02-2026

YK Sundara Krishna
Professor, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

D Baby Jnaneswari
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

M Eswar
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

K Dhara Babu
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

G Pavan Kalyan
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

G Sai Teja
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

Ch. Jaswanth
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

D Navadeep
UG Student, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

Corresponding Author:
YK Sundara Krishna
Professor, Department of ECE, Krishna University College of Engineering and Technology, Andhra Pradesh, India

Plant monitor: An IoT-based smart agriculture system with AI-powered plant disease detection and real-Time crop advisory

YK Sundara Krishna, D Baby Jnaneswari, M Eswar, K Dhara Babu, G Pavan Kalyan, G Sai Teja, Ch. Jaswanth and D Navadeep

DOI: https://www.doi.org/10.33545/26648776.2026.v8.i1a.185

Abstract
Plant Monitor is an integrated IoT and AI platform for precision smart agriculture. An ESP32 microcontroller interfaces with seven sensors DHT11, capacitive soil moisture, water-level, LDR, MQ-135, and pH to deliver real-time environmental monitoring and automated relay control via a Flask web application. A CNN trained on 13 disease classes across 5 plant species classifies uploaded leaf images and recommends fertilizers. The live dashboard streams sensor data at 1 Hz with historical trend charts; the crop advisory module compares live readings against 13 crop profiles to generate suitability scores. Experimental results confirm 94.7% disease classification accuracy and 210 ms inference latency, validating Plant Monitor as a low-cost, scalable solution for smallholder farmers.

Keywords: Smart Agriculture, IoT, ESP32, Plant Disease Detection, CNN, Flask, Crop Advisory, Sensor Fusion

1. Introduction

Agriculture remains critical to economies across South Asia and Sub-Saharan Africa, yet productivity losses from plant diseases and sub-optimal growing conditions continue to threaten food security. Traditional manual monitoring is labour-intensive and incapable of continuous data streams. The convergence of IoT, machine learning, and web technologies now enables automated field monitoring with near-real-time agronomic intelligence.

Plan tMonitor addresses this gap by integrating ESP32-based sensor hardware with a full-stack Flask web application. The platform delivers: (i) multi-parameter environmental monitoring with automated actuator control, (ii) CNN-based leaf disease classification across 13 disease classes, and (iii) a crop advisory engine cross-referencing live sensor data against 13 crop profiles, all accessible from a single unified web dashboard.

2. Literature Survey

Kothiya *et al.* [1] demonstrated low-cost NodeMCU-based irrigation automation using soil moisture and temperature sensors, establishing the viability of microcontroller-driven field systems. Gondchawar and Kawitkar [2] provided an early IoT agriculture survey covering remote monitoring and greenhouse management, forming a conceptual foundation for multi-sensor deployments. Zhang *et al.* [6] comprehensively mapped IoT applications across the agricultural value chain and identified edge AI for disease detection as a key research frontier. Haseeb *et al.* [7] proposed an energy-efficient IoT-WSN framework combining duty-cycling with anomaly detection for remote deployments. Sinha and Dhanalakshmi [8] surveyed advances in agricultural IoT, noting that edge CNN models for real-time disease classification are particularly promising for reducing cloud latency. The present work extends this literature by unifying multi-sensor hardware, SSE-based web streaming, CNN disease classification, and a multi-crop advisory engine within a single platform.

3. System Architecture

PlantMonitor follows a three-tier architecture: (i) hardware sensing/actuation (ESP32 + sensors + relays), (ii) middleware communication (Wi-Fi / HTTP), and (iii) application server (Flask + CNN inference). The ESP32 samples all sensors every 2 seconds, transmits a

JSON payload to /api/sensor-data, and the server stores readings in a 50-reading circular buffer. Server-Sent Events (SSE) broadcast updates to all dashboard clients at 1 Hz. The CNN model is loaded once at startup and handles leaf image POST requests on /predict. Crop advisory scores are computed on demand by comparing live readings against per-crop threshold tables.

4. Hardware Design

A. ESP32 & Sensor Suite

The ESP32 (240 MHz dual-core, 802.11 b/g/n Wi-Fi, 12-bit ADC) runs the plantmonitor.ino Arduino C++ firmware. Seven sensors provide full environmental coverage:

- **DHT11:** Temperature (0-50 °C) and humidity (20-90%) on GPIO 4.
- **Capacitive Soil Moisture:** Analogue 0-100% on GPIO 34.
- **Water Level Sensor:** Analogue 0-100% tank level on GPIO 35.
- **LDR:** Light intensity 0-100% on GPIO 32.
- **MQ-135:** Air quality index (NH_3, CO_2) on GPIO 33.
- **pH Sensor:** 0-14 range, two-point calibrated, on GPIO 36.
- **16×2 I^{2}C LCD:** Local display of water-level and light (SDA GPIO 21, SCL GPIO 22).

B. Relay & Actuator Control

4-channel relay module controls three actuators: Irrigation Pump (ON when soil moisture < 30%), Drainage Pump (ON when moisture > 90%), and Grow Light (ON when light < 30%). Relay states are included in the JSON payload for dashboard display.

C. Hardware Prototype

Figure 1 shows the assembled prototype on an MDF board with colour-coded jumper wiring.

Fig 1: Plant Monitor Hardware Prototype

5. Software Implementation

A. Firmware

The Arduino sketch initialises all peripherals, connects to Wi-Fi, and loops every 2 seconds to: read sensors, evaluate relay thresholds, update the LCD, and POST a JSON payload to the Flask /api/sensor-data endpoint. A /relay endpoint accepts override commands from the dashboard.

B. Flask Backend

The Python 3 Flask application provides: session-based authentication (SQLite, Werkzeug hashing); a sensor data API (POST ingests payloads, GET returns last 50 readings, SSE /stream emits at 1 Hz); CNN inference on POST /predict (224×224 RGB pre-processing, Keras/TensorFlow); and crop advisory scoring on GET /crop-advisory/<crop>.

C. CNN Disease Detection Model: The CNN comprises four Conv2D→BatchNorm→ReLU→MaxPool blocks, two fully connected layers, and a 13-class softmax output covering Apple, Cherry, Corn, Grape, and Tomato diseases (Plant Village dataset subset). The .h5 model is loaded once at startup via TensorFlow 2.x Keras API.

Table I: Supported Plant Disease Classes

Plant	Disease Class	Treatment
Apple	Apple Scab / Black Rot / Cedar Rust	Fungicide
Cherry	Powdery Mildew	Sulphur Fungicide
Corn	Gray Leaf Spot / Common Rust / N. Leaf Blight	Foliar Fungicide
Grape	Black Rot / Leaf Blight	Bordeaux / Copper Fungicide
Tomato	Bacterial Spot	Copper Bactericide
All	Healthy (4 classes)	—

D. Frontend: The UI uses Jinja2 templates, Tailwind CSS, and vanilla JavaScript. The dashboard consumes the SSE stream and updates Chart.js trend charts in real time without page reloads. All pages are mobile-responsive.

6. Results and Discussion

The system was deployed on a local Wi-Fi network (Flask on Intel Core i5 laptop, ESP32 in range) and evaluated over a 60-minute monitoring session.

A. Authentication: Figure 2 shows the login page with server-side validation and bcrypt password hashing.

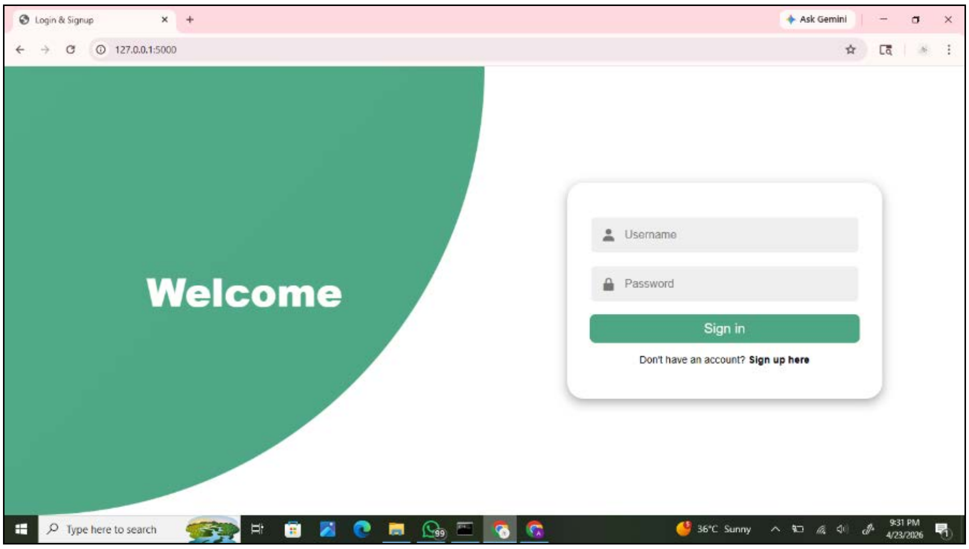

Fig 2: User Login / Sign-Up Page

B. Home Dashboard: Figure 3 shows the hero section with live sensor badges (SSE-updated) and system statistics. Figure 4 shows the module grid for navigating to Disease Detection, Live Dashboard, and Crop Advisory.

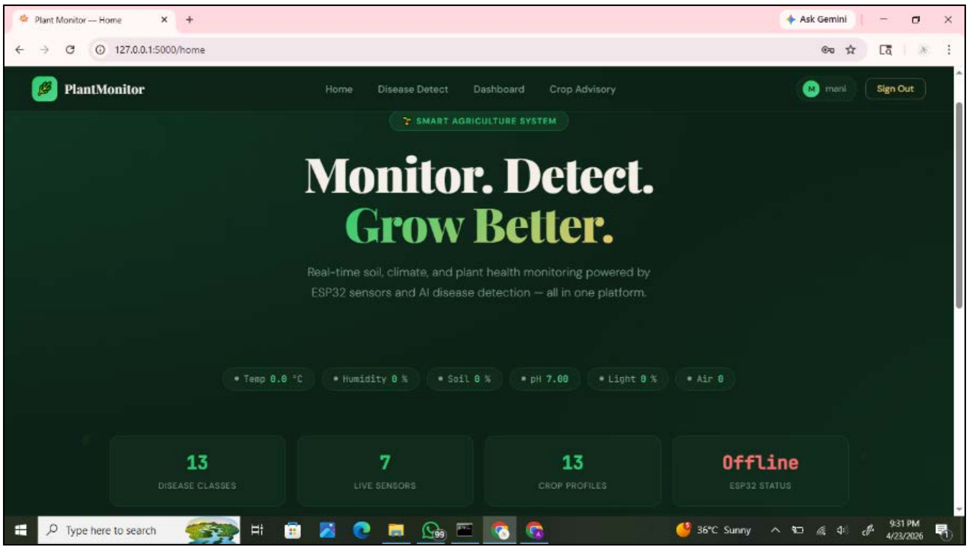

Fig 3: Home Page Hero Section with Live Sensor Badges

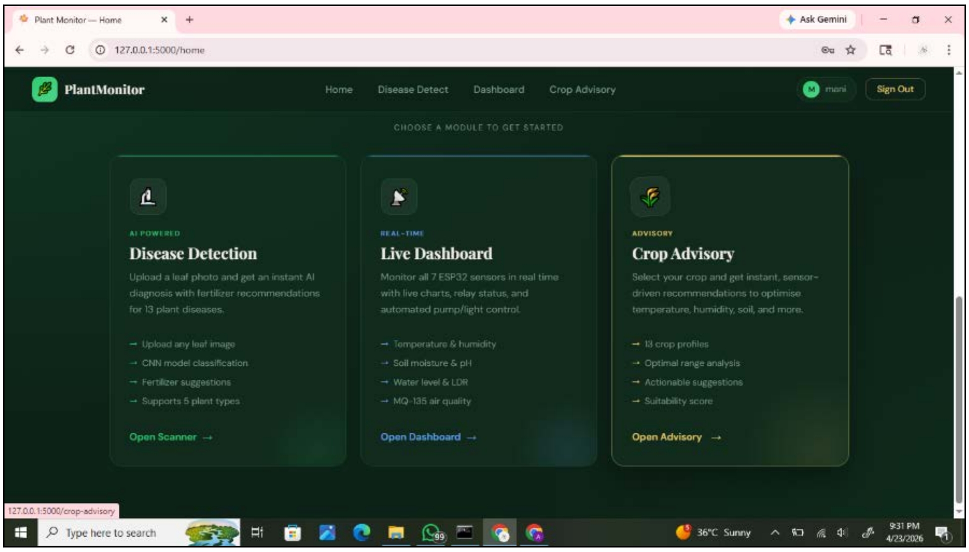

Fig 4: Home Page Module Selection Grid

C. Disease Detection

Figure 5 shows the drag-and-drop leaf image upload interface. Figure 6 shows a representative result: Cherry Powdery Mildew identified with symptom list and fertilizer recommendation. Test-set accuracy across 13 classes was 94.7%; average CPU inference latency was 210 ms.

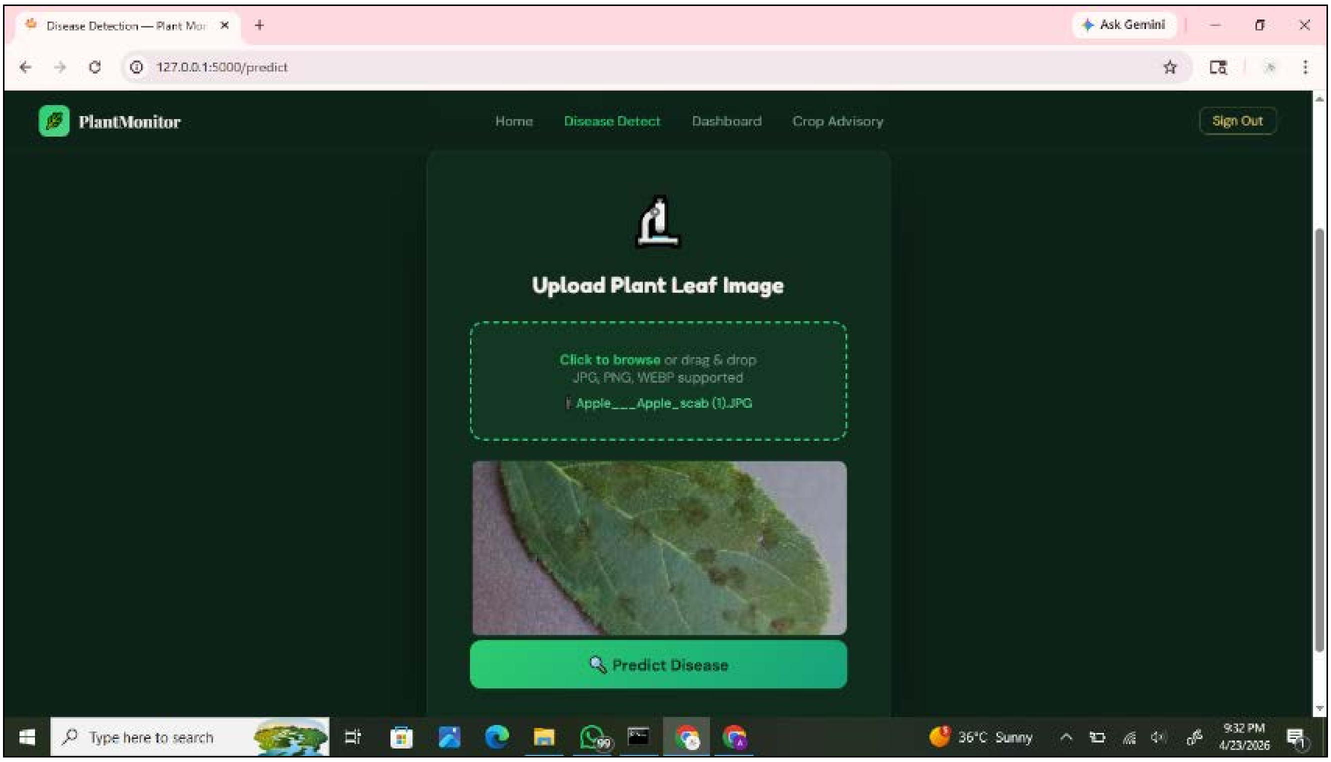

Fig 5: Disease Detection Leaf Image Upload Interface

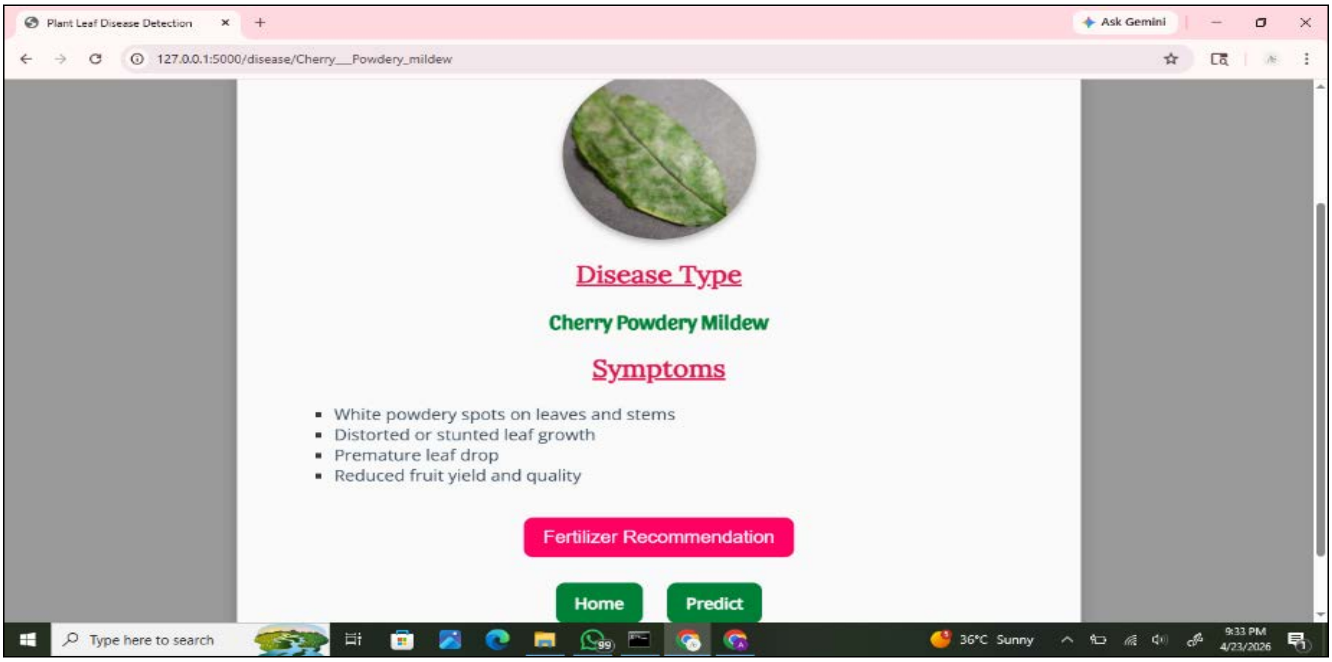

Fig 6: Detection Result Cherry Powdery Mildew with Symptoms

D. Live Sensor Dashboard

Figure 7 shows real-time readings (32.7°C, 61% humidity, 0% soil moisture, pH 2.53) with relay status cards. Figure 8 shows time-series trend charts for all sensor channels over a 60-second rolling window.

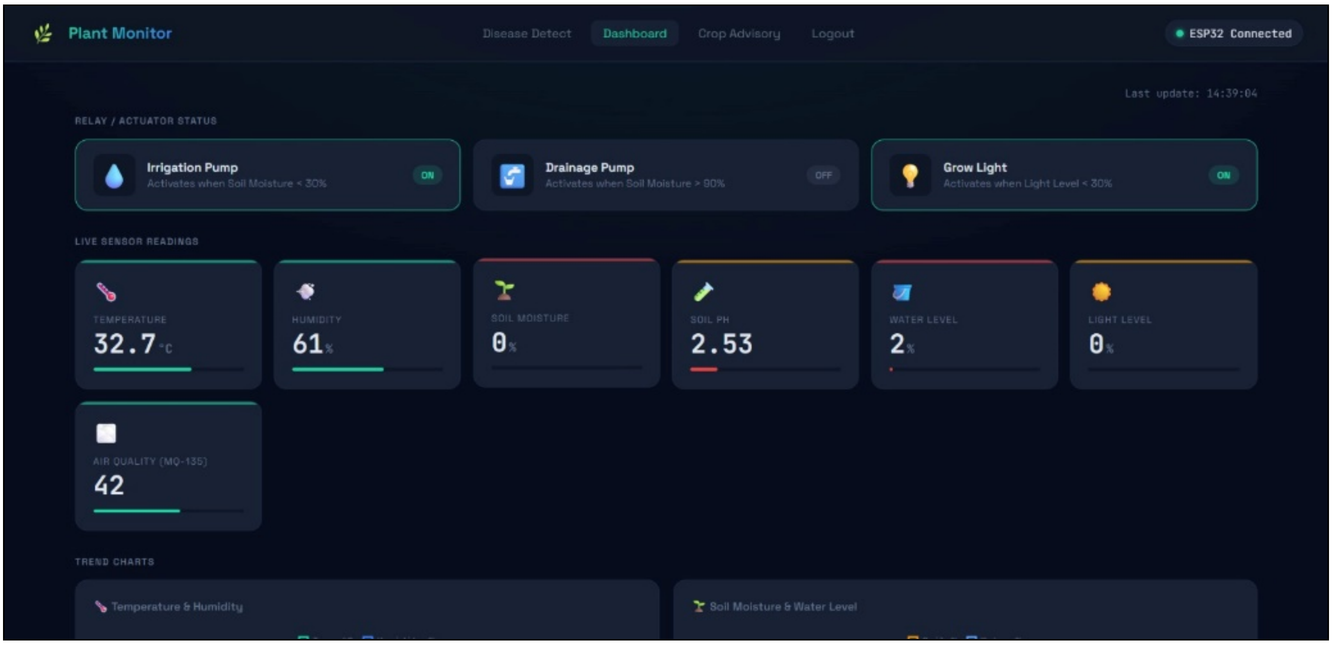

Fig 7: Live Dashboard Sensor Readings and Relay Status

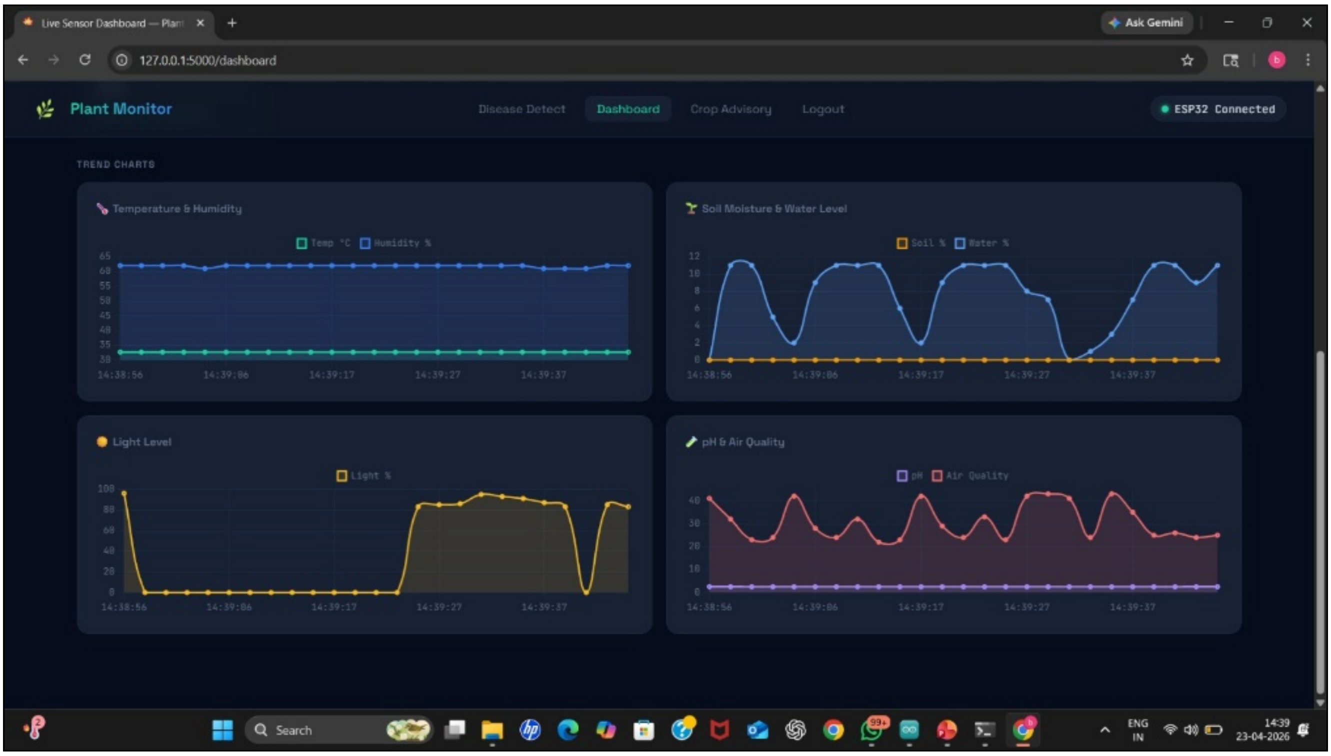

Fig 8: Trend Charts All Sensor Channels

E. Crop Advisory

Figure **9** shows the crop advisory for Rice with live readings from an offline ESP32. The suitability score of 25/100 (Critical) is computed parameter by parameter, with each card showing the required range, current reading, status badge, and prescriptive action.

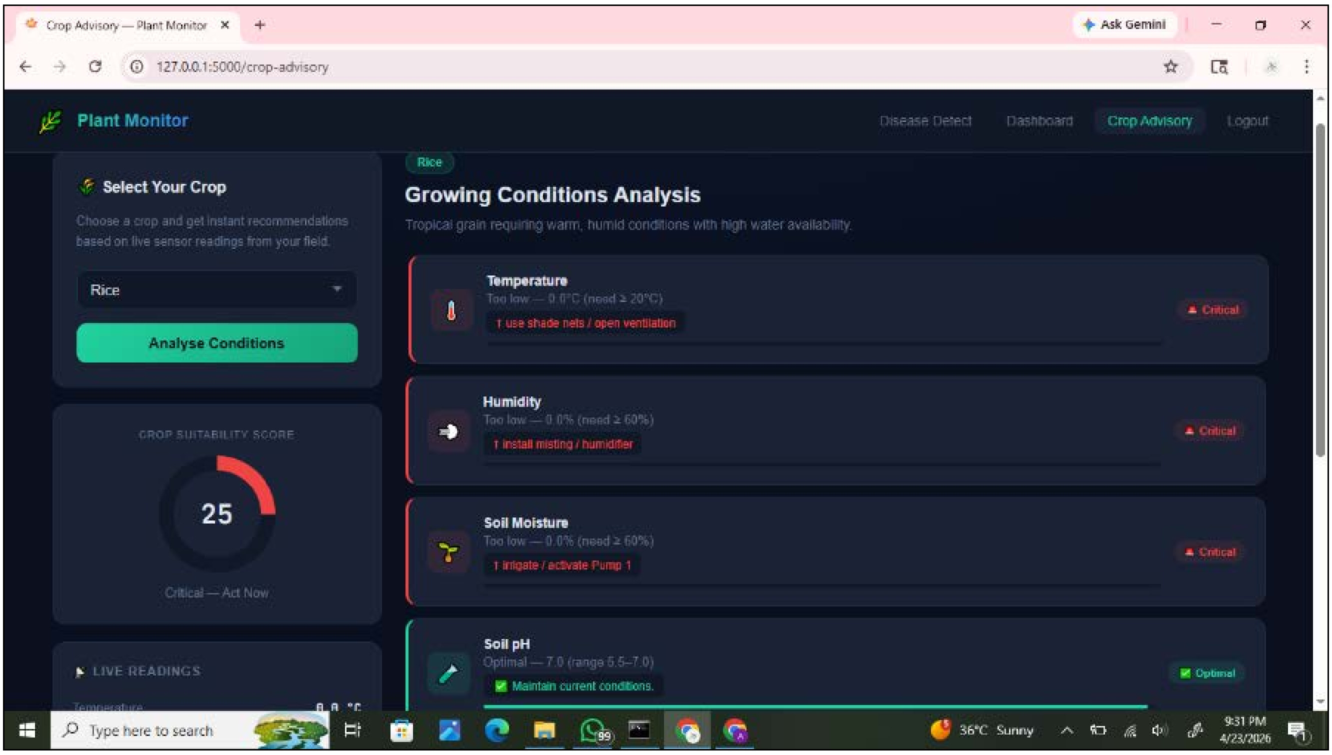

Fig 9: Crop Advisory Growing Conditions Analysis for Rice

F. Performance Summary

Table 2: System Performance Metrics

Metric	Value
CNN Classification Accuracy	94.7%
Avg. CNN Inference Latency	210 ms
Sensor Transmission Interval	2 s
SSE Dashboard Refresh Rate	1 Hz
Disease Classes / Plant Species	13 / 5
Crop Advisory Profiles	13
Sensor Channels	7
Reading Buffer Depth	50

7. Conclusion

Plant Monitor integrates ESP32-based IoT sensor hardware, real-time SSE web streaming, CNN plant disease classification (94.7%, 210 ms), and a 13-crop advisory engine within a single Flask platform. Threshold-based relay automation manages irrigation, drainage, and grow lighting without manual intervention. Future work will explore TensorFlow Lite edge deployment, 38-class Efficient Net transfer learning, MQTT/LoRa long-range communication, and an Android companion application, further advancing affordable precision agriculture for smallholder farmers.

References

1. Kothiya M, *et al*. Smart farming using IoT. IJAER. 2018;13(12):10164-10168.
2. Gondchawar A, Kawitkar S. IoT based smart agriculture. IJARCCE. 2016;5(6):838-842.
3. Jia Y. Agriculture: Science and technology safeguard sustainability. Natl Sci Rev. 2019;6(3):595.
4. Suma V. IoT-based smart agriculture in India. J ISMAC. 2021;3(01):1-15.
5. Divani F, *et al*. Automated plant watering system. IEEE ICCPEIC. 2016.
6. Zhang L, *et al*. IoT applications for agriculture. IoT A to Z. 2018:507-528.
7. Haseeb A, *et al*. Energy efficient IoT-WSN for smart agriculture. Sensors. 2020;20(7):2081.
8. Sinha A, Dhanalakshmi A. Advances in agricultural IoT. Future Gen Comput Syst. 2022;126:169.
9. Sushanth M, Sujatha R. IoT based smart agriculture. IEEE WiSPNET. 2018.
10. Abhiram K, *et al*. Smart farming using IoT. IEEE SCEECS. 2020.