iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85483

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

AgriSense: An AI and IoT-Based Smart Farming System for Crop Disease Detection, Multi-Farm Monitoring and Automated Irrigation

Tamilselvi K[1], Gowtham A[2], Dilip M[3], Blesson P[3]
Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, India

Abstract: ***Crop disease identification, field monitoring and irrigation control are available to farmers today only as separate tools: leaf-disease classifiers are not linked to field data, sensor-based irrigation controllers do not examine crop health, and farmer chatbots have no access to the farm. No single system combines these functions for several farms through one data path. This paper presents AgriSense, an integrated artificial intelligence and Internet of Things (IoT) system for smart farming. A MobileNetV2-based convolutional neural network classifies crop-leaf images submitted through a Flutter mobile application to a FastAPI backend and returns the crop, the detected condition and a confidence value, together with preventive measures, pesticide information and watering advice. An ESP32 controller reads soil-moisture, temperature, humidity and pH sensors, uploads the readings under a farm identifier and drives a relay-operated water pump with a hysteresis rule, with manual and timed operation as alternatives. A farm dashboard with scan history, alerts and a chatbot complete the application. The architecture, algorithms and prototype implementation are described, and a reproducible evaluation plan is given.***

Keywords: Smart Agriculture, Crop Disease Detection, MobileNetV2, ESP32, Automated Irrigation, Internet of Things (IoT), Multi-Farm Monitoring, Mobile Application.

I. INTRODUCTION

Agriculture is the basis of food production, yet field management is still largely manual. A farmer walks the field to inspect the crop, judges from experience whether a leaf spot is a disease, and switches the pump on and off by hand. Late identification of a crop disease affects crop health, watering without measuring the soil wastes water, and supervising several farms in person makes it difficult to compare their conditions or to keep a record of earlier treatments.

Each part of this problem has been addressed, but only by a separate solution. Convolutional neural networks (CNNs) classify leaf diseases on smartphones [5], [9], but are not linked to field sensors or to irrigation. Sensor-driven irrigation systems operate the pump from soil measurements [4], [10], [11], but do not examine crop health. Farmer chatbots answer queries on weather, plant protection and prices [2], [3], but have no access to the farm's own data. A farmer who needs all three must use separate tools whose data are not shared, so the leaf image, the soil reading and the pump action are never considered together.

This paper presents AgriSense, a unified *AI and IoT-based smart farming system* in which one farm identifier connects the leaf image, the soil readings and the pump events of every farm. An *Internet of Things (IoT)* field unit built around an ESP32 senses the soil and weather conditions and controls a relay-driven pump, a MobileNetV2-based service performs *crop disease detection* with a confidence check, and a Flutter mobile application provides *multi-farm monitoring*, preventive and pesticide advice, irrigation control, scan history, alerts and an agricultural chatbot.

II. LITERATURE SURVEY

1) M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E.-H. M. Aggoune, "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk", IEEE Access (2019). This survey reviews wireless sensors, communication protocols, smartphones and cloud computing used in IoT-based agriculture and their application to soil sampling, irrigation, crop disease and pest management. It gives the technological foundation of smart farming but does not present a working system that combines disease detection with irrigation control. [1]

2) M. Momaya, A. Khanna, J. Sadavarte and M. Sankhe, "Krushi – The Farmer Chatbot", ICCICT (2021). An AI chatbot answers farmers' queries on weather, plant protection, market price, fertilizer use and soil testing. It shows that a chatbot makes agricultural advice accessible, but it works on text only and is not linked to field sensors or leaf images. [2]

©IJRASET: All Rights are Reserved

3) B. Arora, D. S. Chaudhary, M. Satsangi, M. Yadav, L. Singh and P. S. Sudhish, "Agribot: A Natural Language Generative Neural Networks Engine for Agricultural Applications", IC3A (2020). A sequence-to-sequence conversational system answers frequently asked farming questions and emphasises disease detection and weather forecasting. Its answers come from a trained question set and are not connected to real-time farm data or irrigation hardware. [3]

4) J. E. Tamil Malar and M. Vaishnavi, "IoT based Smart Irrigation System using ESP32", ICESC (2022). The low-cost ESP32 with built-in Wi-Fi operates irrigation automatically from sensor readings. The study concentrates on irrigation alone and has no disease detection or multi-farm management. [4]

5) 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", IEEE Access (2024). A lightweight CNN based on a modified MobileNetV3Small is deployed in an Android application and reports about 97% precision, recall and accuracy. It proves that mobile-friendly CNNs give accurate in-field diagnosis, but it is restricted to one crop and has no sensing, irrigation or recommendation module. [5]

6) R. Rashid, W. Aslam, R. Aziz and G. Aldehim, "An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models", IEEE Access (2024). IoT data acquisition is combined with several deep learning models for early detection of corn leaf diseases. It shows the benefit of joining sensing with image analysis, but it concerns a single crop and has no irrigation action or farmer-facing advisory interface. [6]

7) J. Samuvel Joseph and G. Srinitya, "Deep Learning Based Rice Plant Disease Detection and Classification System with Automated Treatment Recommendation Engine", ICAAIC (2025). A rice-disease classifier is coupled with a treatment recommendation engine so that the output includes suggested action. The single-crop scope and the absence of live sensing and irrigation control limit the system. [7]

8) J. Kolli, D. M. Vamsi and V. M. Manikandan, "Plant Disease Detection using Convolutional Neural Network", IEEE IBSSC (2021). A CNN classifies plant leaf images into healthy and diseased categories and discusses the effect of data preparation on accuracy. It remains a classification study without deployment on a mobile application or integration with farm data. [8]

9) M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", IEEE/CVF CVPR (2018). MobileNetV2 introduces inverted residual blocks with linear bottlenecks, giving a lightweight and accurate network for mobile and embedded vision. It is the architecture selected for AgriSense, but it must be fine-tuned with a crop-disease dataset before agricultural use. [9]

10) Y. Kim, R. G. Evans and W. M. Iversen, "Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network", IEEE Transactions on Instrumentation and Measurement (2008). A distributed wireless sensor network with a supervisory control unit senses and controls irrigation remotely. It establishes the value of sensor-based irrigation over manual operation but uses dedicated field networks and provides no mobile application or disease analysis. [10]

11) 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", IEEE Transactions on Instrumentation and Measurement (2014). Soil-moisture and temperature sensors feed a microcontroller-based gateway that uses a GPRS module to operate irrigation valves. It shows that automated irrigation conserves water but is restricted to irrigation and has no image-based crop-health analysis. [11]

12) S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing", ICCUBEA (2015). An image-processing approach performs segmentation, feature extraction and classification of leaf images. It is an early automated method whose hand-crafted features are less robust than deep learning under varied lighting and backgrounds. [12]

13) V. Balafas, E. Karantoumanis, M. Louta and N. Ploskas, "Machine Learning and Deep Learning for Plant Disease Classification and Detection", IEEE Access (2023). The review compares machine-learning and deep-learning techniques and datasets for plant disease classification, highlights the superiority of deep CNNs and transfer learning and notes the need for practical deployment. It does not itself propose an integrated system. [13]

14) Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing", Proceedings of the IEEE (2019). The survey discusses running AI models at the network edge to reduce latency and cloud dependence. It motivates the on-device inference planned for AgriSense but gives no agriculture-specific implementation. [14]

15) C. Suresh, S. Srikaran, T. Senthilkumar, V. Magizhan and S. Santhosh, "An Offline Mobile Application for Plant-Disease Detection Using MobileNetV3 and TensorFlow Lite", ICCVBIC (2026). An offline smartphone application runs a MobileNetV3 model through TensorFlow Lite without an internet connection. It supports on-device diagnosis in the field but is limited to disease detection, with no irrigation control or multi-farm management. [15]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

III. RESEARCH GAP

The surveyed works each provide one or two of the capabilities that a farm requires, as summarized in Table I. The following gaps remain.

A. Separate Solutions without Integration

Disease detection, treatment advice, field sensing, irrigation control and farmer assistance are offered by different systems that do not share data. No surveyed work provides all of them through one data path.

B. Restriction to a Single Crop or a Single Function

Most disease-detection studies address one crop [5], [6], [7], and irrigation studies address watering alone [4], [10], [11]; the results therefore cannot support a farmer who grows several crops on several plots.

C. No Farm-Wise Record

No surveyed work keeps, for each farm, a single record of leaf scans, sensor readings and pump events that the farmer can review, compare and act on.

D. Uncertain Predictions Presented as Fact

Classifiers usually return a label for every image; a poor photograph then produces a confident but unreliable diagnosis. A confidence check that asks for a clearer image is rarely reported.

TABLE I.
CAPABILITIES OF RELATED WORK AND AGRISENSE (YES = PROVIDED; PART. = PARTIALLY; – = NOT DESCRIBED)

Work	Disease detection	Treatment advice	Live sensing	Irrigation control	Mobile app	Multi-farm
[1]	Part.	–	Part.	Part.	–	–
[2], [3]	–	Part.	–	–	Part.	–
[4], [10], [11]	–	–	Yes	Yes	–	–
[5], [15]	Yes	–	–	–	Yes	–
[6]	Yes	–	Yes	–	–	–
[7]	Yes	Yes	–	–	–	–
[8], [9], [13]	Yes	–	–	–	–	–
AgriSense	Yes	Yes	Yes	Yes	Yes	Yes

IV. PROBLEM STATEMENT

Farmers lack a unified, low-cost system that, through a single farm identifier, (i) identifies crop diseases from a leaf image and reports how reliable the result is, (ii) gives preventive, pesticide and watering advice, (iii) measures the soil and weather conditions of each farm and operates the water pump from the measurements, and (iv) keeps a farm-wise record of scans, readings and irrigation that can be reviewed from one mobile application.

V. OBJECTIVES

1) To develop an IoT field unit that measures soil moisture, temperature, humidity and pH and controls a water pump through a relay.
2) To develop an AI service that identifies the crop and the disease from a leaf image and reports a confidence value.
3) To provide preventive measures, pesticide information and watering advice for the detected condition.
4) To operate irrigation automatically with a hysteresis rule and manually or by timer from the mobile application.
5) To store scans, readings and pump events under a farm identifier so that several farms can be managed from one account.
6) To build a mobile application with a dashboard, history, alerts and an agricultural chatbot, and to define a reproducible evaluation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. METHODOLOGY

A. System Architecture

AgriSense has two cooperating workflows that meet at the farm dashboard, as shown in Fig. 1. In the AI-based disease analysis workflow, the farmer logs in or registers, adds or selects a farm or zone, and captures or uploads a leaf image. The image is sent to the FastAPI backend, which pre-processes it and passes it to the MobileNetV2 model; the detected condition is classified, treatment, pesticide and watering recommendations are generated, and the analysis is saved to the farm history. In the IoT-based monitoring and irrigation workflow, the farm sensors measure temperature, humidity, soil moisture and pH, the ESP32 controller reads them and checks whether the soil moisture is low, and the data are sent to the backend and stored under the farm identifier. If the moisture is low, the relay is switched ON and the water pump starts irrigation; otherwise the pump stays OFF and monitoring continues.

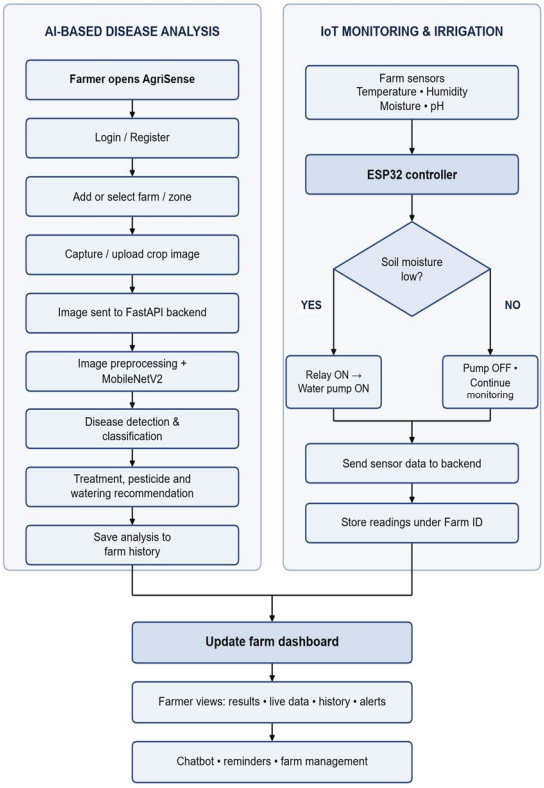


Fig. 1. Architecture of the AgriSense system: AI-based disease analysis (left) and IoT-based monitoring and irrigation (right) meeting at the farm dashboard.

B. Field Unit Hardware

The field unit follows a sense–process–control structure (Fig. 2). Agricultural sensors measure soil moisture, temperature, humidity and soil pH. An ESP32 microcontroller with built-in Wi-Fi [16] reads the sensors, communicates with the backend and drives a relay. The relay isolates the low-voltage controller from the DC water pump, which is powered from a separate supply and is never connected to a controller pin. The electronics are housed in a retrofit enclosure with an antenna.

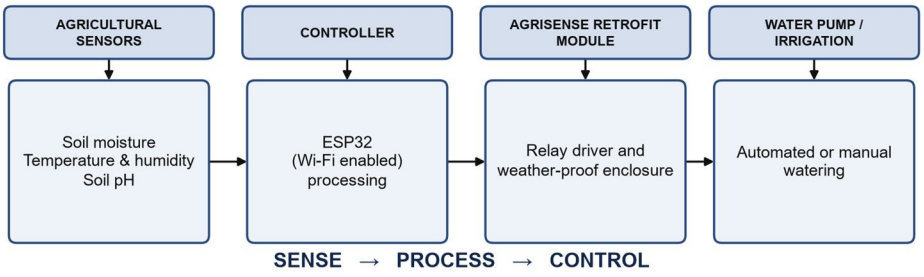


Fig. 2. Hardware architecture of the AgriSense field unit (sense – process – control).

C. Backend and Database

A FastAPI server is the communication layer between the mobile application, the AI model, the IoT unit and a relational database. It validates payloads, runs inference and exposes REST endpoints (Table II). The database holds five main tables (Table III); every record carries a farm identifier, so several farms are managed from one account. Passwords are stored only as salted hashes, and requests carry a signed token with an expiry time.

TABLE II.
PRINCIPAL REST ENDPOINTS

Method	Endpoint	Purpose
POST	/predict	Disease detection from a leaf image
POST	/sensor-data	Store a sensor reading
GET	/sensor-data	Latest reading of a farm
POST	/pump/control	Manual pump command
GET	/history	Detection history of a farm
GET	/farm	Farms of a user

TABLE III.
DATABASE TABLES

Table	Principal fields
users	user_id, name, email, phone, password hash
farms	farm_id, user_id, name, location, crop, zone
sensor_readings	farm_id, temperature, humidity, soil_moisture, soil_ph, time
detections	farm_id, crop, disease, confidence, image reference, time
pump_events	farm_id, command, reason (manual / automatic), time

D. AI-Based Crop Disease Detection

The classifier uses MobileNetV2 [9], whose inverted residual blocks with linear bottlenecks give high accuracy at small model size, which suits a service handling many requests and later conversion for on-device use [14], [15]. A network pre-trained on ImageNet is the feature extractor; its head is replaced by global average pooling, a dropout layer and a softmax layer with one output per crop–condition class (for example Tomato–Early Blight, Potato–Healthy and Pepper–Bacterial Spot). Fine-tuning proceeds in two stages (Table IV): the head is trained with the base frozen, then the top layers are trained with a small learning rate. Data augmentation by rotation, flipping and brightness change improves robustness to field conditions, and the labelled dataset is divided into training, validation and test subsets. The processing pipeline is shown in Fig. 3.

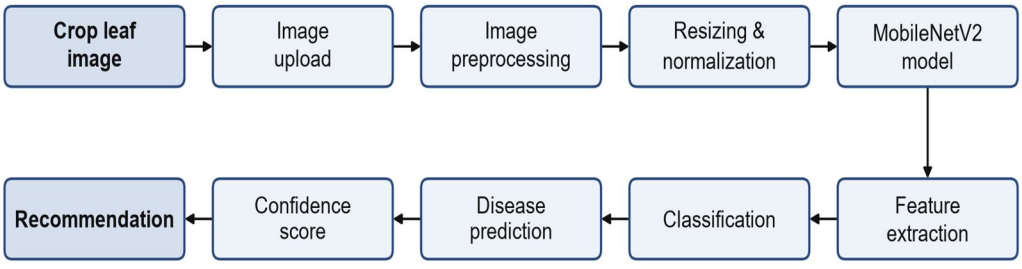


Fig. 3. AI processing pipeline for crop disease detection.

TABLE IV.
FINE-TUNING CONFIGURATION

Parameter	Value
Input size	224 × 224 × 3
Backbone	MobileNetV2, ImageNet weights
Head	Global average pooling, dropout 0.3, softmax
Stage 1	Head only, Adam, 10 epochs
Stage 2	Top 30 layers, learning rate 1e-5, 5 epochs
Augmentation	Rotation, flip, brightness
Confidence threshold τ	60 %

For an uploaded image the backend converts it to RGB, resizes it to 224 × 224 pixels, scales the pixel values to [0, 1] and obtains the class probabilities p_k. The predicted class and its confidence are

$$K = \mathit{argmax}\ p_k, \quad C = 100 \cdot p_K \tag{1}$$

If C is below the threshold τ, the service asks for a clearer image instead of reporting a disease, which avoids presenting an uncertain diagnosis as fact. Otherwise it looks up preventive measures, pesticide information and watering advice for the predicted class in a knowledge table, which is kept as data so that it can be reviewed by agronomists without retraining, and stores the result in the farm history. The advice is decision support and does not replace an agricultural expert.

E. Soil Sensing and Irrigation Control

The ESP32 reads the soil-moisture sensor through its analogue-to-digital converter. With calibration readings R_{dry} in dry soil and R_{wet} in saturated soil, the moisture percentage is

$$M = 100 \cdot (R_{dry} - R) / (R_{dry} - R_{wet}), \quad 0 \le M \le 100 \tag{2}$$

Temperature, humidity and pH are read from their sensors, and the four values are posted as a JSON document to the backend at a fixed interval with the farm identifier. In automatic mode the pump follows a hysteresis rule with a lower threshold M_{low} and an upper threshold M_{high}:

$$pump = ON \text{ if } M < M_{low}; \quad OFF \text{ if } M \ge M_{high} \tag{3}$$

Two thresholds prevent rapid switching around a single value. Temperature and humidity guards stop the pump from starting in hot or very humid conditions, and a longest-single-run limit forces it off after a fixed time. The defaults are listed in Table V. The first prototype used a temperature-only rule, which was enough to demonstrate relay switching; the final system replaces it with the soil-moisture rule, as compared in Fig. 4, and keeps temperature as a guard and a monitored parameter.

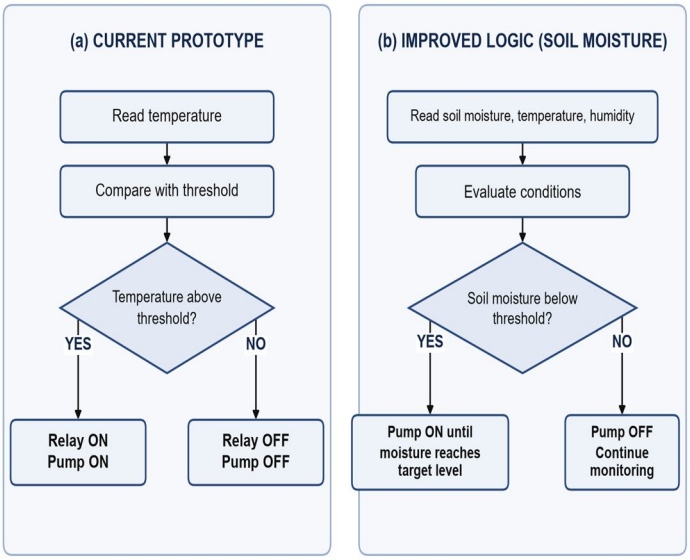


Fig. 4. Irrigation logic: (a) first prototype using temperature and (b) final soil-moisture-based logic.

TABLE V.
IRRIGATION PARAMETERS OF THE PROTOTYPE

Parameter	Default
Start moisture M_low	30 %
Stop moisture M_high	60 %
Temperature guard	40 °C
Humidity guard	90 %
Longest single run	30 min
Run timers	10, 20, 30, 60 min

The farmer can also start or stop the pump from the application, choose a run timer or schedule a run. A manual command is stored by the backend and fetched by the ESP32 in its next communication cycle, every start and stop is recorded with its reason, and the application states that changes take effect when an offline controller reconnects.

F. Mobile Application and Chatbot

The Flutter application is the farmer's only interface. A bottom navigation bar gives access to Home, Farms, Camera, Chat and Settings. Home highlights scans that need attention and shows the weather; Farms lists the farms and zones and opens a detail page with the crop, area, sensor hub and scan history; Camera captures or selects a leaf image; and the irrigation page controls the pump. The farm assistant answers questions on diseases, watering and crop care, and reminders notify the farmer of pending tasks such as an irrigation check or a follow-up scan.

VII. RESULTS AND DISCUSSION

A. Prototype Implementation

The complete workflow was exercised on the prototype application. Fig. 5 shows the home dashboard, the Scan a Leaf screen with camera and gallery options and recent scans, and the irrigation page with automatic mode, timers, guards and irrigation history. Table VI lists the functions that were exercised and the behaviour observed.

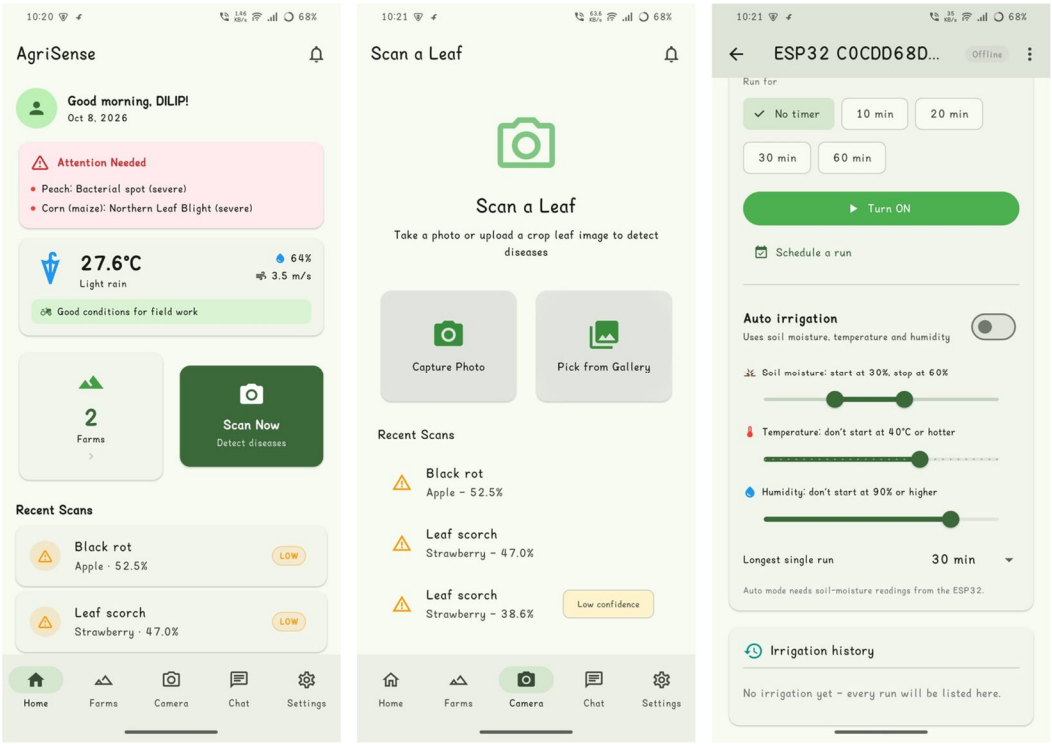


Fig. 5. Prototype screens: home dashboard, Scan a Leaf and irrigation settings.

TABLE VI.
FUNCTIONS EXERCISED ON THE PROTOTYPE

Function	Observed behaviour
Farm list	Farms listed with crop, acres and location
Farm details	Crop, acres, zone, location and sensor hub
Leaf scan	Camera or gallery input; recent scans listed
Scan history	Crop, condition, confidence and date stored
Low confidence	Uncertain scans tagged instead of confirmed
Sensor hub	Temperature and humidity shown; offline state flagged
Irrigation	Manual run, timers, schedule and automatic mode
Assistant	Chat on crop diseases and farm data
Settings	Account, language, theme and server address

B. Evaluation Plan

Because the project is at the prototype stage, a formal protocol is defined rather than reporting unmeasured figures. The classifier will be assessed on a held-out test set that was not used in training or validation, with the metrics in Table VII, where TP, TN, FP and FN are the numbers of true and false positives and negatives for a class:

$$Acc = (TP + TN) / (TP + TN + FP + FN), \quad F1 = 2 \cdot P \cdot R / (P + R) \tag{4}$$

where P = TP / (TP + FP) is the precision and R = TP / (TP + FN) is the recall. Water saving will be measured by comparing the volume pumped under the soil-moisture rule with that under a fixed schedule on adjacent plots, and reliability by the fraction of scheduled sensor uploads that reach the server.

TABLE VII.
PLANNED EVALUATION METRICS

Aspect	Metric	Method
Classifier	Accuracy, P, R, F1	Held-out test set
Image quality	Rejection rate	Poor-light and blurred images
Latency	Time per scan	Upload to result
Irrigation	Water volume	Rule versus fixed schedule
Sensing	Calibration error	Gravimetric reference
Reliability	Upload success	Uptime over a week

C. Discussion

The prototype shows that functions currently offered by separate tools can be provided together through one farm identifier: the leaf image, the soil reading and the pump action of a farm are stored and presented in one place. Two design choices make the system cautious. The classifier refuses to label an image when its confidence is low, and the controller uses hysteresis, guards and a run limit so that a faulty reading cannot flood the field or wear the pump by rapid switching. The advice is stored as data, so agronomists can review it independently of the software. The evaluation is limited to the prototype: classification accuracy and water saving are not yet measured, sensors need calibration for each soil, different crops need different thresholds, and cloud connectivity is currently required. The advice must be validated by agricultural expertise.

VIII. CONCLUSION AND FUTURE SCOPE

AgriSense unifies MobileNetV2-based crop disease detection, preventive and pesticide advice, ESP32-based sensing and relay irrigation, multi-farm management and a chatbot, functions that existing solutions provide separately, in one low-cost mobile platform. The farm identifier keeps images, readings and pump events organised for every farm, and the confidence check and the hysteresis control make the system cautious in both diagnosis and watering. Future work includes the quantitative evaluation of the classifier and of water saving described above, weather integration, additional soil and water-level sensors, on-device inference with TensorFlow Lite [14], [15], GPS-based farm maps, regional-language support, encrypted device communication and a solar-powered field unit.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

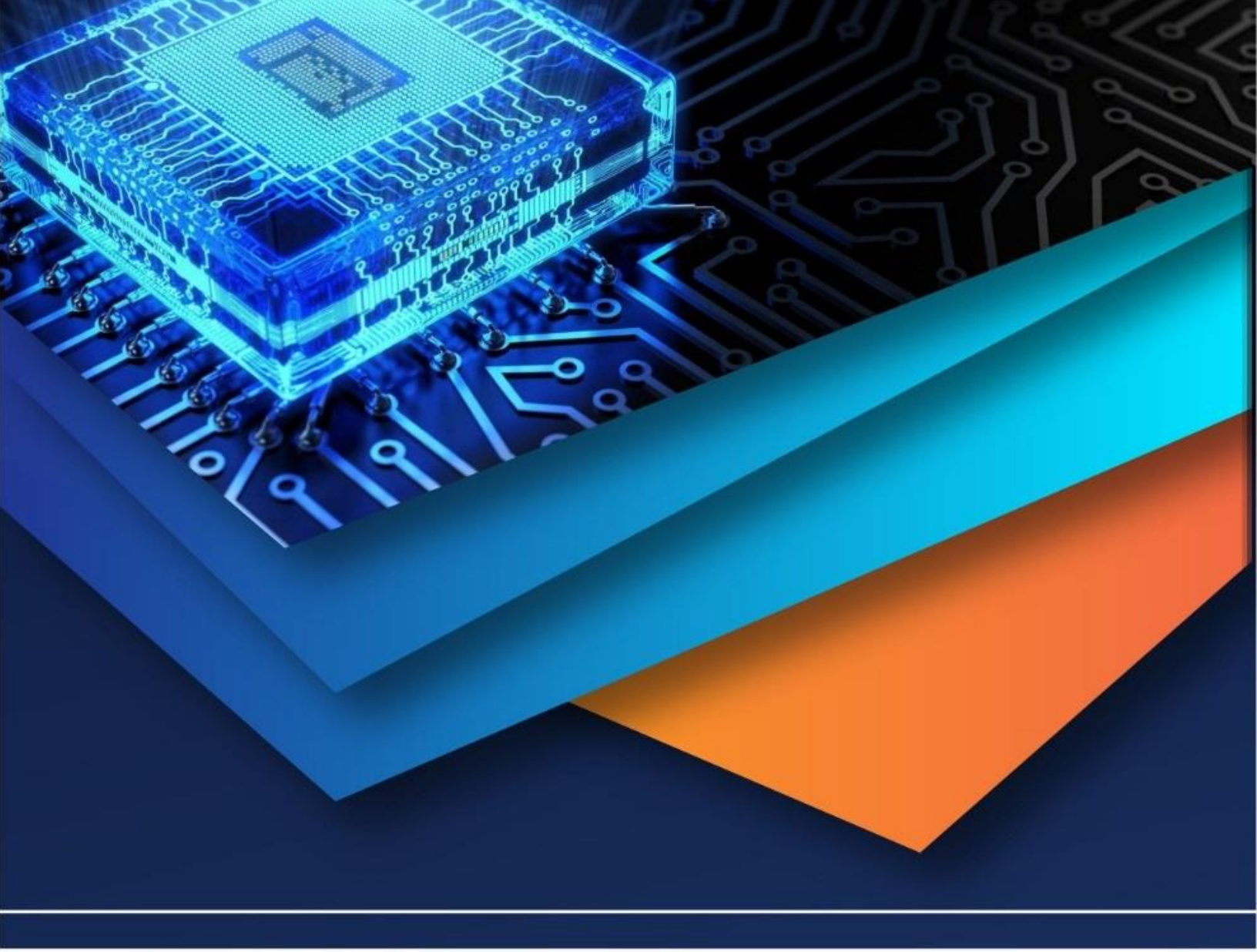
IX. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, for institutional support.

REFERENCES

[1] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, "Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk," IEEE Access, vol. 7, pp. 129551–129583, 2019.
[2] M. Momaya, A. Khanna, J. Sadavarte, and M. Sankhe, "Krushi – The farmer chatbot," in Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT), 2021, pp. 1–6.
[3] B. Arora, D. S. Chaudhary, M. Satsangi, M. Yadav, L. Singh, and P. S. Sudhish, "Agribot: A natural language generative neural networks engine for agricultural applications," in Proc. Int. Conf. Contemp. Comput. Appl. (IC3A), 2020, pp. 28–33.
[4] J. E. Tamil Malar and M. Vaishnavi, "IoT based smart irrigation system using ESP32," in Proc. 3rd Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), 2022, pp. 1751–1755.
[5] 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," IEEE Access, vol. 12, pp. 83575–83588, 2024.
[6] R. Rashid, W. Aslam, R. Aziz, and G. Aldehim, "An early and smart detection of corn plant leaf diseases using IoT and deep learning multi-models," IEEE Access, vol. 12, pp. 23149–23162, 2024.
[7] J. Samuvel Joseph and G. Srinitya, "Deep learning based rice plant disease detection and classification system with automated treatment recommendation engine," in Proc. 4th Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC), 2025, pp. 198–204.
[8] J. Kolli, D. M. Vamsi, and V. M. Manikandan, "Plant disease detection using convolutional neural network," in Proc. IEEE Bombay Sect. Signature Conf. (IBSSC), 2021, pp. 1–6.
[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.
[10] Y. Kim, R. G. Evans, and W. M. Iversen, "Remote sensing and control of an irrigation system using a distributed wireless sensor network," IEEE Trans. Instrum. Meas., vol. 57, no. 7, pp. 1379–1387, Jul. 2008.
[11] 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," IEEE Trans. Instrum. Meas., vol. 63, no. 1, pp. 166–176, Jan. 2014.
[12] S. D. Khirade and A. B. Patil, "Plant disease detection using image processing," in Proc. Int. Conf. Comput. Commun. Control Autom. (ICCUBEA), 2015, pp. 768–771.
[13] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas, "Machine learning and deep learning for plant disease classification and detection," IEEE Access, vol. 11, pp. 114352–114377, 2023.
[14] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," Proc. IEEE, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
[15] C. Suresh, S. Srikaran, T. Senthilkumar, V. Magizhan, and S. Santhosh, "An offline mobile application for plant-disease detection using MobileNetV3 and TensorFlow Lite," in Proc. 7th Int. Conf. Comput. Vis. Bio Inspired Comput. (ICCVBIC), 2026, pp. 1463–1469.
[16] Espressif Systems, "ESP32 series datasheet." [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)