



IoT-Based Smart Farm Protection and AI Crop Disease Detection System

Dr. C. P. Divate¹, H. M. Mangure², N. R. Bhokare³, Pranav Narendra Narkar⁴,
Gaytri Suresh Nandiwale⁵, Sakshi Chandrakant Mayane⁶, Sandesh Sanjay
Suryawanshi⁷

¹Dean, Project Guide, Department of Computer Engineering, Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India.

²Project Guide, Department of Computer Engineering, Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India.

³HOD, Department of Computer Engineering, Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India.

^{4,5,6,7}Department of Computer Engineering, Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India.

To Cite this Article: Dr. C. P. Divate¹, H. M. Mangure², N. R. Bhokare³, Pranav Narendra Narkar⁴, Gaytri Suresh Nandiwale⁵, Sakshi Chandrakant Mayane⁶, Sandesh Sanjay Suryawanshi⁷, "IoT-Based Smart Farm Protection and AI Crop Disease Detection System", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 476-480.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](#); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Crop damage caused by unwanted animals and the difficulty of identifying crop diseases at an early stage are two common problems faced by farmers. This paper proposes the IoT-Based Smart Farm Protection and AI Crop Disease Detection System, a low-cost smart agriculture system that addresses both problems through a single web platform. For farm protection, a copper wire arrangement installed around the farm boundary acts as the sensing element; when an animal touches it, an ESP32/Arduino-class microcontroller detects the contact and activates a buzzer, and the farmer can turn the buzzer OFF remotely through the website. For crop health monitoring, a camera captures an image of a crop leaf, and an AI/ML-based image classification model identifies the possible disease, which is displayed on the website. The proposed classifier is a lightweight convolutional neural network (MobileNetV2, transfer learning) to be trained on the public PlantVillage leaf image dataset. The web application provides homepage, admin and farmer modules, and a MySQL database stores user, farm and crop details, intrusion alerts, buzzer control records and disease detection results. The system is organized into six modules: homepage, admin login, farmer login, crop disease detection, farm protection and database design. This paper presents the proposed architecture, methodology and planned evaluation; the system is at the design stage, all modules are described as proposed, and no experimental results are claimed.

Key Word: Smart Agriculture, Internet of Things (IoT), Animal Intrusion Detection, Crop Disease Detection, Artificial Intelligence, Image Classification, MobileNetV2, ESP32, Flask, MySQL, Web Application

I. INTRODUCTION

Agriculture plays an important role in the rural economy, but that directly affect crop production. One of the major problems is crop damage caused by unwanted animals entering agricultural fields. Another is the difficulty of identifying crop diseases at an early stage, because traditional identification depends mainly on manual observation, which is time-consuming and may not be early or accurate.

The Internet of Things (IoT) and Artificial Intelligence (AI) offer a practical way to reduce these problems. Microcontrollers and sensing arrangements can detect events on the farm and raise an alert, cameras can capture leaf images, and machine-learning models can classify the images to identify possible diseases. Recent studies combine IoT, sensors, machine learning, image processing and web platforms for plant disease detection, irrigation management and remote monitoring of agricultural fields [1]–[4].

This paper proposes the IoT-Based Smart Farm Protection and AI Crop Disease Detection System. A copper wire arrangement installed around the farm boundary detects animal contact, and the IoT controller activates a buzzer to alert the farmer. The farmer can monitor the system and turn the buzzer OFF remotely through a website. In addition, a camera captures an image of a crop leaf, which is analyzed using an AI/ML-based image classification approach, and the possible disease is displayed on the website. The web platform, named AgriShield AI, is supported by a MySQL database that stores users, farm and crop details, alerts and detection results.

The main contributions of this work are: (i) a proposed design that combines IoT-based farm protection and AI-based crop disease detection in a single system built from low-cost components; (ii) a description of the sensing circuit, controller logic and communication flow of the copper-wire intrusion alert with remote buzzer control; (iii) a web platform with admin and farmer roles, alert and detection history, and a database design that supports them; and (iv) a planned evaluation covering intrusion detection, buzzer response time, disease detection accuracy, communication reliability and website performance. The scope is a prototype to be tested with a boundary wire in an agricultural field and with crop leaf

images; automated irrigation and additional sensors are left for future work.

The rest of the paper is organized as follows. Section II reviews the literature. Section III describes the methodology as six modules. Section IV discusses the expected benefits, planned evaluation and feasibility. Section V concludes.

II. LITERATURE REVIEW

Ibrahim et al. [1] proposed an AI-IoT based smart agriculture system for plant disease detection and treatment. The study addresses plant diseases, pest attacks and water shortage by combining AI with IoT, using a hardware-based agricultural pivot together with image processing and deep learning. A dataset of 25,940 images was used to classify 11 classes of plant leaves with a pre-trained ResNet50 model, which reached a testing accuracy of 99.8%. A mobile application gives farmers guidance on disease detection and treatment. The work shows that AI, image processing and IoT can be integrated for automated plant disease detection.

Gorijavolu et al. [2] proposed an intelligent irrigation system that combines IoT and machine learning for early detection of sugarcane leaf diseases. Soil temperature, humidity and leaf characteristics such as texture and colour changes are collected from IoT-connected sensors and from high-resolution UAV images. A Non-linear Growing Self-Organizing Map (NG-SOM) inside an Artificial Neural Network (ANN) improves feature extraction and classification. For diseases such as red rot, smut and rust, the approach is reported to reach a detection accuracy of 95.6%.

Tace et al. [3] developed a smart irrigation system based on IoT and machine learning. Soil humidity, temperature and rain sensors collect real-time data, which is managed with the Node-RED platform and a MongoDB database. KNN, Logistic Regression, Neural Networks, SVM and Naïve Bayes were compared, and KNN gave the best recognition rate of 98.3% with an RMSE of 0.12. A web application lets users visualize the sensor data and supervise the agricultural environment.

Adenugba et al. [4] proposed a solar-powered smart irrigation system using an Internet of Everything (IoE) approach. Sensors and environmental data monitor water level, weather, humidity, temperature and irrigation information, and a Radial Basis Function Network (RBFN) predicts environmental conditions to support irrigation control. A web platform allows authorized users to monitor and control the system remotely, which reduces unnecessary water use.

Research gap. The reviewed studies show that IoT, sensors, machine learning and web platforms are effective in agriculture, but they concentrate on irrigation management, environmental monitoring and plant disease detection. Comparatively little attention is given to protecting farms from animal intrusion and to a simple remote control mechanism for the protection system, and none of the reviewed systems brings intrusion alerts, remote buzzer control and leaf disease detection together on one platform. The proposed work does not claim a new learning technique. Its main objective is to integrate three functions in a single low-cost web platform: (a) a copper-wire farm intrusion alert, (b) remote buzzer OFF control for the farmer, and (c) AI-based crop disease detection. The prototype is planned to be tested for detection accuracy, response time and communication reliability.

III. METHODOLOGY

A. System Overview

The proposed system follows a modular pipeline (Fig. 1). In the farm protection path, the copper wire arrangement and the ESP32 controller detect animal contact and drive the buzzer, and the event is sent over Wi-Fi to a Flask backend server. In the disease detection path, a camera captures a crop leaf image, which the backend pre-processes with OpenCV and classifies with an AI/ML model. A MySQL database stores all records, and the web application shows alerts, buzzer status and disease results to the farmer and the administrator. The system is organized into six modules (Table I), which are described in Sections III-B to III-E in the order of their numbers.

[Insert Fig. 1 block diagram here]

Fig. 1. Block diagram of the proposed IoT-Based Smart Farm Protection and AI Crop Disease Detection System.

No.	Name	Function	Key tools
1	Homepage	Home, about, how it works, disease detection, farm protection, contact, login, FAQs	HTML, CSS, JavaScript
2	Admin login	Dashboard, users, alerts, buzzer, disease records, crop information, reports	Flask, MySQL
3	Farmer login	Profile, farm monitoring, alerts, remote buzzer control, disease results, history	Flask, MySQL
4	Crop disease detection	Leaf image capture or upload, pre-processing, AI/ML classification, result display	Camera, OpenCV, Python, ML
5	Farm protection	Boundary wire detection, buzzer activation, alert generation, remote OFF	ESP32/Arduino, copper wire, buzzer
6	Database design	Users, farmers, crops, alerts, buzzer control, detection, disease information, notifications	MySQL

Table 1 summary of the six modules

B. Modules 1–3: Web Application

The web application is developed with HTML, CSS and JavaScript on the front end and Python/Flask on the back end. The homepage (Module 1) presents the Home, About Us, How It Works, Disease Detection, Farm Protection, Contact Us, Login and FAQs sections. The admin module (Module 2) provides a dashboard, profile update, user management, farm protection and buzzer management, disease detection management, crop information management, reports and analytics, notifications, database management, password change and system settings. The farmer module (Module 3) provides profile management, farm monitoring, animal intrusion alerts, buzzer status, remote buzzer control, crop disease detection, detection results and history, and notifications. Passwords are stored in hashed form, and access to each module depends on the user role.

C. Module 4: Crop Disease Detection

The farmer captures a crop leaf image with the connected camera or uploads a previously captured image. The image is pre-processed with OpenCV, and an AI/ML-based classification model implemented in Python analyzes the visual characteristics of the leaf and identifies the possible disease. The result is displayed through the web interface, and the image reference, disease name and detection date are stored so that the farmer can view the detection history. A disease information table gives the basic symptoms and prevention information for each disease.

Proposed model and dataset. The proposed classifier is MobileNetV2, a lightweight convolutional neural network, used with transfer learning: the network is initialised with ImageNet weights and its final layers are fine-tuned on leaf images. A lightweight model is chosen because it can run on an ordinary computer or low-cost server without a GPU. The proposed training dataset is the public PlantVillage dataset, which contains roughly 54,000 labelled leaf images in 38 healthy and diseased classes. Only the classes of the crops grown on the target farm will be selected, and the final list of crops and disease classes will be confirmed when the dataset is prepared. Leaf images collected from the farm are planned as an additional test set to check performance on real field images.

Processing pipeline (proposed). Each image is resized to 224×224 pixels, normalised, and passed to the model, which returns the most probable class and a confidence score. The dataset is planned to be split into training, validation and test sets (for example 80%/10%/10%), and light augmentation (rotation, flip, brightness change) is planned to improve robustness. The model type, dataset and class list are design choices at this stage; no training has been carried out and no accuracy is reported.

D. Module 5: Farm Protection

Sensing circuit (proposed). Two bare copper wires are run in parallel around the farm boundary at a small gap, so that an animal touching them bridges the gap. One wire (the sense wire) is connected to a digital input pin of the ESP32 that is configured as INPUT_PULLUP, so the pin normally reads HIGH through the internal pull-up resistor. The second wire is connected to the common ground (GND). When an animal touches both wires, the sense wire is pulled to ground and the pin reads LOW, which the firmware treats as an intrusion event. The sense wire carries only the 3.3 V logic level, and a series resistor (for example 1 kΩ) is placed between the wire and the pin to limit current and protect the input. The buzzer is connected to a digital output pin, through a transistor driver if its current exceeds the pin rating, with its other terminal to ground. Table II lists the connections.

Component	ESP32 pin	Connection / purpose
Copper sense wire	Digital input (INPUT_PULLUP)	Via series resistor; normally HIGH, LOW on contact
Copper ground wire	GND	Common ground; contact with the sense wire pulls the pin LOW
Buzzer	Digital output	HIGH = ON, LOW = OFF (transistor driver if needed)
Wi-Fi	Built-in	HTTP requests to the Flask server

Table II proposed controller connections

[Insert circuit diagram here: copper wires, ESP32, resistor, buzzer]

Operation. The firmware reads the sense pin continuously. When the pin goes LOW and stays LOW for a short debounce time (for example 50 ms), the controller (1) sets the buzzer output ON, giving an immediate local alert, and (2) sends an HTTP POST request over Wi-Fi to the Flask backend with the device identifier and event time. The backend stores an intrusion alert with its detection time and buzzer status and shows it on the farmer's dashboard. The controller also requests the current buzzer command from the backend at a fixed interval (for example every two seconds). When the farmer presses the buzzer OFF button on the website, the backend records the control action with its time and source (website or IoT system), and the controller sets the buzzer output OFF at its next request. Previous alerts are kept as alert history for monitoring and

analysis. The wire arrangement, pin numbers and timing values above are proposed design values and will be fixed during implementation.

E. Module 6: Database Design

The MySQL database is built around eight tables (Table III). Users are identified as Admin or Farmer by the role field. Farmers are linked to their crops, intrusion alerts, buzzer control records, disease detection records and notifications through farmer_id, and disease detection records are also linked to crops through crop_id. The disease information table provides reference data for the diseases identified by the AI/ML model.

Table	Key fields
Users	user_id, name, mobile_number, email, password (hashed), role, created_at
Farmers	farmer_id, user_id, farm_name, location, crop_type, contact_number
Crops	crop_id, farmer_id, crop_name, variety, planting_date, status
Intrusion alerts	alert_id, farmer_id, detection_time, alert_type, buzzer_status, alert_status
Buzzer control	control_id, farmer_id, buzzer_status, control_time, control_source
Disease detection	detection_id, farmer_id, crop_id, image_path, disease_name, detection_result, detection_date
Disease information	disease_id, disease_name, crop_name, symptoms, prevention
Notifications	notification_id, farmer_id, notification_type, message, status, created_at

Table III database tables and key fields

F. Tools and Facilities

The prototype is planned with an ESP32/Arduino microcontroller board, copper wire, a buzzer, connecting wires and a camera module or web camera. Software tools are Visual Studio Code and the Arduino IDE, HTML, CSS, JavaScript, Python, Flask, OpenCV, TensorFlow/Keras (or a similar machine-learning library) and MySQL. Testing uses web browsers with developer tools, Postman for backend API testing, and the Serial Monitor for the IoT hardware. Development and testing are carried out in the college computer and project laboratories, and the farm protection and leaf-image tests are planned in an agricultural field; Wi-Fi or internet connectivity is needed for communication between the IoT device, the server and the web platform.

IV. EXPECTED BENEFITS AND EVALUATION PLAN

A. Expected Benefits

Based on the design and the literature, the system is expected to: give an immediate alert when an animal touches the boundary wire, which can help reduce crop damage; let the farmer turn the buzzer OFF remotely, reducing the need to be physically present near the farm; identify possible crop diseases from leaf images more quickly than manual inspection, supporting early action; reduce manual monitoring through one web platform for alerts, buzzer status and disease results; and offer an affordable, user-friendly solution that encourages IoT and AI in farming. These are design goals, not measured results.

B. Planned Evaluation

After implementation, the system will be evaluated on the following measures. Results will be reported in a Results and Discussion section once testing is complete.

- animal intrusion detection accuracy and alert generation rate;
- buzzer response time after animal contact, and the response time and reliability of the remote buzzer OFF control;
- disease detection accuracy (including per-class precision and recall) and the image processing time needed to produce a result;
- website response time and IoT communication reliability between the sensors, controller, server and web platform;
- system uptime and the accuracy of sensor readings and stored data;
- user satisfaction, collected as feedback from farmers on usability and convenience.

C. Feasibility and Scope

The project is technically feasible because it uses commonly available technologies: ESP32/Arduino boards, copper wire, a buzzer, a camera, Python, Flask, OpenCV and MySQL. It is economically feasible because the prototype needs only low-cost hardware and open-source software, and operationally feasible because the web interface is designed for farmers with varying levels of technical skill. For use beyond the laboratory, better camera modules, a computer with enough processing power for training the AI/ML model, cloud hosting (for example AWS or Azure), secure authentication and regular database backup would be required. The limitations are that the boundary wire detects only animals that touch it, the disease detection depends on the quality and coverage of the leaf image dataset, the system depends on Wi-Fi or internet connectivity, and it does not yet include automated irrigation or additional sensors.

V. CONCLUSION

This paper proposed the IoT-Based Smart Farm Protection and AI Crop Disease Detection System, which combines a copper-wire and buzzer based animal intrusion alert, remote buzzer control through a website, camera-based leaf image capture, AI/ML-based crop disease identification and a MySQL-backed web platform for farmers and administrators. The six-module design lets each part be developed, tested and improved independently, and it uses low-cost hardware and widely available software, so it can serve as a practical prototype for smart agriculture.

Future work includes completing the implementation, evaluating the system as planned in Section IV, and adding more sensors, automated irrigation and advanced disease detection models.

Acknowledgment

The authors thank their project guides, Dr. C. P. Divate and Mr. H. M. Mangure, the Head of Department, Ms. N. R. Bhokare, and the Department of Computer Engineering, Sanjay Bhokare Group of Institute, Miraj, for their guidance and support.

REFERENCES

1. A. S. Ibrahim, S. Mohsen, I. M. Selim, R. Alroobaea, M. Alsafyani, A. M. Baqasah, and M. Eassa, "AI-IoT based smart agriculture pivot for plant diseases detection and treatment," [journal name, volume, article number, DOI – to be completed from the published article], 2025.
2. D. Gorijavolu, K. Sharma, and N. S. Rao, "Smart irrigation system and early plant disease detection using IoT and novel non-linear growing self-organizing map based artificial neural network," *Sci. Rep.*, 2026, doi: 10.1038/s41598-025-33323-w.
3. Y. Tace, M. Tabaa, S. Elfilali, C. Leghris, H. Bensag, and E. Renault, "Smart irrigation system based on IoT and machine learning," *Energy Rep.*, [vol., pages, DOI – to be completed], Nov. 2022.
4. F. Adenugba, S. Misra, R. Maskeliūnas, R. Damaševičius, and E. Kazanavičius, "Smart irrigation system for environmental sustainability in Africa: An Internet of Everything (IoE) approach," *Math. Biosci. Eng.*, vol. 16, no. 5, pp. 5490–5503, 2019, doi: 10.3934/mbe.2019273.