International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

SMART CROP DISEASE PREDICTION USING ARTIFICIAL INTELLIGENCE AND IoT

Thoutam Maneesha and N. Sukumar
Department of Electronics and Communication Engineering,
Jayamukhi Institute of Technological Sciences, Warangal, Telangana, India

Abstract—Agricultural productivity is increasingly affected by crop diseases and environmental variability. Conventional disease identification relies heavily on periodic visual inspection, which can be subjective and difficult to scale. This paper presents an Artificial Intelligence of Things (AIoT)-based crop-health monitoring and disease-prediction framework that combines environmental sensing, camera-based leaf-image acquisition, embedded processing, wireless communication, and artificial-intelligence models. An ESP32-based node acquires temperature, relative humidity, soil-moisture, and light-intensity measurements, together with crop or leaf images. Sensor observations are checked, filtered, range-validated, and normalized, while images are resized, denoised, converted to RGB, and normalized before analysis. A convolutional neural network (CNN) is used for visual disease classification, and Random Forest and XGBoost models are considered for environmental-data classification. Image and sensor predictions are combined through a weighted decision-fusion stage to generate crop-health risk information and farmer-facing alerts. Wi-Fi communication using HTTP or MQTT supports cloud transmission, while local storage and edge processing provide a fallback during connectivity interruptions. This paper reports 92% field classification accuracy across 19 significant crop-disease categories and an 18–23% improvement over compared monocausal models. Finally the increase in farmer recommendation compliance from 54% to 88%, and daily active application usage of 94% compared with 31% for traditional agricultural applications

Index Terms—AIoT; crop disease prediction; Internet of Things; ESP32; convolutional neural network; Random Forest; XGBoost; sensor fusion; smart agriculture.

1. INTRODUCTION

Agriculture is central to food security, economic development, and rural livelihoods. Crop diseases can reduce yield and quality when they are not recognized and managed promptly. Conventional diagnosis commonly depends on farmers or agricultural experts inspecting plants and interpreting visible symptoms. This process can be time-consuming, subjective, and difficult to maintain continuously over large fields. Symptoms may also be subtle during early disease development, while environmental stress can complicate visual assessment.

Recent advances in artificial intelligence (AI), the Internet of Things (IoT), computer vision, and cloud/edge computing enable connected agricultural systems to collect information continuously and convert it into actionable insight. IoT sensors can measure temperature, humidity, soil moisture, and light intensity. Camera modules can capture leaf images, and AI models can analyze the resulting visual and numerical data. The combination of AI and IoT, commonly described as AIoT, therefore provides an opportunity to move from periodic inspection towards continuous, context-aware crop monitoring.

Image-based deep learning has demonstrated potential for plant disease recognition. Mohanty et al. [1], Brahimi et al. [2], Sladojevic et al. [12], and Ferentinos [5] investigated deep-learning approaches for recognizing disease symptoms from plant images. However, image-only systems may not account for the environmental conditions in which symptoms develop. Temperature, humidity, soil-water condition, and sunlight can provide complementary context for crop-health assessment.

This work presents an AIoT architecture that integrates environmental sensing, leaf-image analysis, embedded acquisition, wireless communication, cloud/edge processing, and decision support. The ESP32 is used as the central sensing and communication controller. A CNN analyzes crop images, while Random Forest and XGBoost are considered for numerical sensor data. The outputs are combined to estimate disease risk and communicate alerts through a mobile- or web-oriented interface.

The principal problem addressed is the development of a cost-conscious, real-time crop-monitoring platform that can acquire both environmental and visual information, analyze these data using AI, and provide timely disease-risk information to farmers. Systems that rely only on manual inspection require human effort and may not scale. Systems that rely only on environmental sensing may not observe visual symptoms, while image-only systems may omit the context associated with crop stress and disease susceptibility.

The motivation is to combine complementary data sources in one modular system. Temperature and humidity

sensors characterize atmospheric conditions; a soil-moisture sensor measures water conditions around the plant; a light sensor measures illumination; and a camera provides visual evidence from crop or leaf surfaces. The ESP32 provides a low-cost embedded platform with wireless connectivity. Edge processing can support local operation when Internet connectivity is weak, while cloud services can store historical measurements and support centralized analysis.

The primary objectives of the paper are

- Develop an AIoT crop-monitoring node to acquire temperature, humidity, soil-moisture, and light-intensity measurements together with crop/leaf images.
- Integrate AI models for visual disease classification and environmental-data analysis.
- Combine visual and environmental predictions to generate disease-risk information and alerts.
- Support wireless transmission, cloud/local storage, and farmer-oriented visualization.

II. RELATED WORK

Deep learning has been applied to plant disease recognition through automated feature extraction from leaf images. Mohanty et al. [1] explored deep learning for image-based plant disease detection. Brahimi et al. [2] studied tomato disease classification and symptom visualization, while Sladojevic et al. [12] demonstrated deep neural-network recognition of plant diseases from leaf images. Ferentinos [5] evaluated deep-learning models for plant disease diagnosis. These studies establish the relevance of CNN-based visual analysis but are primarily image-oriented.

Kamilaris and Prenafeta-Boldú [3] surveyed deep learning applications in agriculture, including image analysis, crop monitoring, and disease detection. Their review highlights the broad applicability of learned representations to agricultural data. The practical performance of image-based models, however, can be affected by illumination, background, crop variety, disease stage, and field conditions. Image preprocessing and representative training data are therefore important.

IoT-based smart agriculture systems focus on distributed sensing and communication. Nisha and Ramya [11], Ahamed and Pathan [15], Deshmukh and Sawarkar [17], and Kumar et al. [21] discuss connected sensing architectures for agricultural monitoring. Patil and Kale [13] consider soil monitoring using IoT and NPK sensors. Such systems provide continuous measurements but do not necessarily perform image-based disease diagnosis.

Research on intelligent crop protection has explored the combination of connected sensing and machine learning. Zhang et al. [4] describe an IoT-based approach to real-time vegetable disease detection using machine learning. Other project-cited works address AI and IoT integration [14], crop-disease prediction [18]–[20], and real-time crop-health monitoring [16], [23], [24]. Collectively, the literature motivates integrated systems that connect sensing, classification, and farmer decision support.

The reviewed approaches commonly emphasize one or more of image-based disease classification, sensor-based monitoring, or smart-farming connectivity. The project identifies an opportunity to combine these functions in a unified platform. The proposed framework therefore integrates environmental sensing, camera-based image acquisition, ESP32 communication, AI classification, cloud/edge processing, and a farmer-facing alert mechanism. This integration is intended to provide both visual evidence and environmental context for crop-health assessment.

III. PROPOSED SYSTEM AND METHODOLOGY

The proposed system is organized as a sequence of sensing, preprocessing, communication, AI inference, multimodal fusion, and decision-support stages. The architecture is modular: individual sensors, image acquisition, communication, and AI models can be developed and tested independently before integration.

At the sensing layer, environmental sensors measure temperature, relative humidity, soil moisture, and light intensity. A camera module captures crop or leaf images. The ESP32 reads sensor values, coordinates image capture where supported by the hardware configuration, attaches timestamps, performs basic validation, and manages communication. The computational layer preprocesses the data and supplies them to the AI models. The communication layer transmits information to cloud or local infrastructure using Wi-Fi with HTTP or MQTT. The intelligent-action layer combines model outputs and generates alerts and recommendations.

3.1 Data Acquisition

The ESP32 periodically reads the connected sensors and initiates image acquisition. Sensor observations and images are associated with a timestamp; geo-tagging can be included when GPS information is available. The acquisition interval and transmission frequency can be configured according to the crop, power budget, and

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

communication conditions. The source report does not specify a fixed sampling interval or a calibrated sensor model for all measurements.

3.2 Data Preprocessing

Preprocessing is used to reduce invalid observations and make the data suitable for model inference. The sensor pipeline includes missing-value checking, noise filtering, range validation, and normalization. Invalid or out-of-range values should be flagged rather than silently treated as valid measurements. Normalization is applied so that features with different units and numerical ranges can be supplied consistently to the sensor model.

3.2.1 Image Preprocessing

The project implementation specifies resizing images to 224 × 224 pixels, applying Gaussian smoothing with a 3 × 3 kernel, converting BGR images to RGB, and scaling pixel values to the interval [0, 1]. The preprocessing operation can be expressed as:

$$I_norm = RGB(GaussianBlur(Resize(I, 224 \times 224))) / 255.$$

The project methodology also discusses contrast/brightness normalization and background filtering to address variation in field illumination and background. These operations should be applied consistently during model training and inference.

3.3 Wireless Communication and Data Management

The ESP32 uses Wi-Fi as the primary communication mechanism. HTTP requests can transfer sensor readings and image information to a server, while MQTT can support lightweight publish/subscribe messaging. Firebase is identified as a possible cloud database for synchronizing observations and predictions with the user interface. When Internet connectivity is unavailable, the architecture supports temporary local storage followed by transmission when the connection is restored.

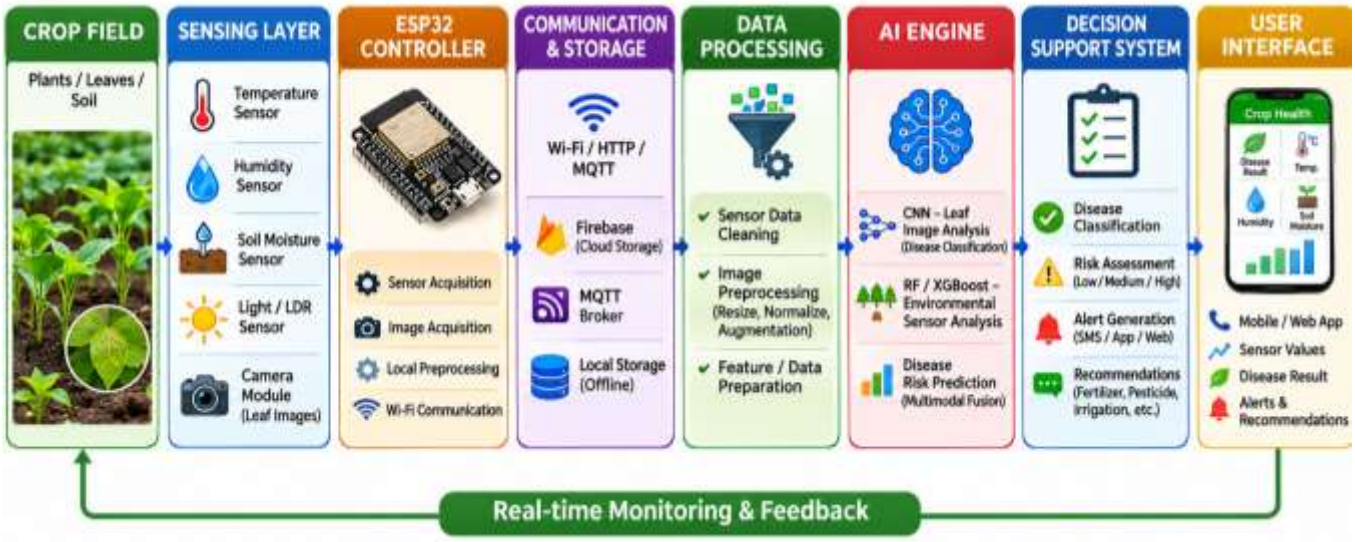


Figure 1. Functional architecture of the proposed AIoT crop-disease prediction system.

3.4 Cloud and Edge Processing

Edge processing places initial processing close to the data source and is useful when network connectivity is weak or intermittent. The ESP32 can perform acquisition, validation, and basic preprocessing; more computationally intensive inference may be placed on a nearby edge device depending on hardware capability. Cloud processing provides centralized storage, historical analysis, model management, and scalability. The design therefore uses cloud and edge resources as complementary components rather than assuming continuous cloud availability.

3.5 CNN-Based Image Classification

A convolutional neural network is used to learn discriminative visual features from crop/leaf images. The source implementation defines convolutional blocks with 32, 64, and 128 filters, 3 × 3 kernels, ReLU activation, batch normalization, and max-pooling. Global average pooling, dense classification, and dropout are included in the model structure. The implementation specifies an image size of 224 × 224, batch size of 32, and up to 20 training epochs, with early stopping used during training.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

$$Y(i,j) = \Sigma_m \Sigma_n X(i+m, j+n) K(m,n) + b,$$

where X is the input feature map, K is the convolution kernel, b is the bias, and Y is the resulting feature map. For C output classes, the softmax probability is:

$$P_i = exp(z_i) / \Sigma_{(j=1)}^{C} exp(z_j), \quad \hat{y} = arg\ max_i\ P_i.$$

The disease label associated with the highest predicted probability is returned as the image-model output. The healthy class and disease labels associated with apple, corn, grape, potato, and tomato. The list contains 19 labels in total, including Healthy; therefore, it should not be interpreted as 19 disease classes plus a healthy class.

3.6 Environmental-Data Classification

Numerical environmental measurements are supplied to machine-learning classifiers. The paper identifies Random Forest and XGBoost as the candidate models. The sensor-data pipeline includes data loading, missing-value handling, invalid-record removal, label encoding, and feature scaling. The trained model and scaler are saved for subsequent inference. The source code expects a sensor-data file containing the environmental features and a corresponding class label; the report does not provide the full dataset schema, sample count, or class distribution.

Random Forest combines predictions from an ensemble of decision trees, while XGBoost builds an additive sequence of trees to reduce prediction error. The paper uses these models as alternative or complementary classifiers for the numerical sensor stream. Model selection should be based on validation performance and should avoid data leakage between training and test observations.

3.7 Multimodal Decision Fusion

The image model provides visual evidence of disease, while the sensor model provides environmental context. The two outputs are combined using a weighted fusion score:

$$S = \alpha\ S_{image} + (1 - \alpha)\ S_{sensor}, \quad 0 \leq \alpha \leq 1,$$

where S_image is the image-derived risk score, S_sensor is the sensor-derived risk score, α is a weighting coefficient, and S is the fused score. The weighting coefficient should be selected using validation data and fixed before final testing. If model outputs are not calibrated probabilities, calibration should be performed before combining them.

A rule-based decision engine converts the fused score and model agreement into an alert level. When image and sensor outputs both indicate a potential problem, the system can assign a higher-priority warning. If the models disagree or confidence is below a selected threshold, the interface should communicate uncertainty and request additional inspection rather than presenting an unsupported definitive diagnosis.

3.8 Farmer Interface and Alert Generation

The decision-support module presents sensor readings, image classification, disease-risk information, and recommendations through a mobile- or web-based interface. The paper describes accessibility features such as color coding, pictorial instructions, vibration patterns, and voice messages, with low-bandwidth operation as a design consideration. The interface should distinguish model predictions from expert-confirmed diagnoses and should communicate that recommendations may require confirmation by an agricultural specialist.

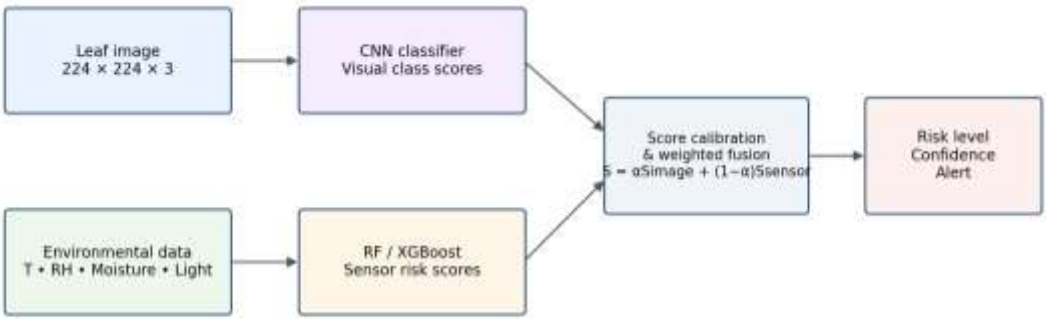


Figure 2. Data flow for sensor/image transmission through Wi-Fi using HTTP or MQTT to cloud/local storage.

3.10 Training and Evaluation Procedure

The general model-development workflow consists of dataset collection, cleaning, preprocessing, training/testing partitioning, model training, validation/tuning, final evaluation, and deployment. Image and sensor datasets should

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

be split independently while preserving class representation. Where multiple images or sensor records originate from the same plant, plot, or acquisition session, grouping should be used to prevent related observations from appearing in both training and test sets.

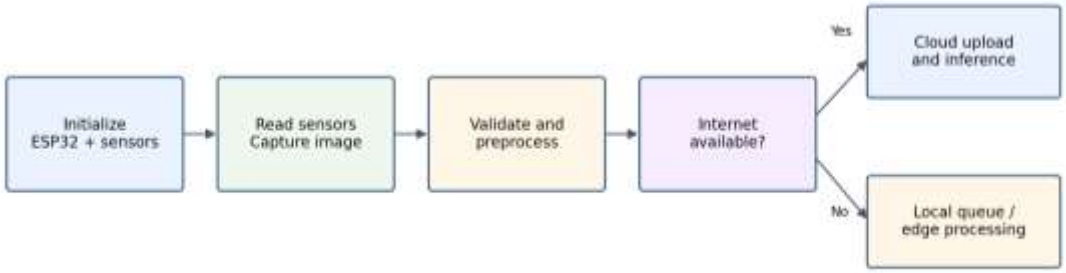


Fig 3. Embedded deployment workflow with cloud/edge connectivity handling.

4. EXPERIMENTAL SETUP AND RESULTS

The paper describes a greenhouse verification stage followed by a real-field pilot. The experimental platform comprises an ESP32 controller, temperature and humidity sensing, a soil-moisture sensor, an LDR/light sensor, an ESP32-CAM or compatible camera, Wi-Fi connectivity, a power supply/battery, and cloud/local storage. Arduino IDE is identified for firmware development; Wi-Fi and HTTPClient libraries support communication; MQTT and Firebase are identified for messaging and data storage. Python and a CNN framework are used for AI development, with Random Forest and XGBoost for environmental-data classification.

Table I. Hardware and software configuration

Layer/component	Specified platform or function
Controller	ESP32; sensor interfacing and wireless communication
Environmental sensing	Temperature, relative humidity, soil moisture, and light intensity
Visual acquisition	ESP32-CAM/compatible camera; crop and leaf images
Communication	Wi-Fi; HTTP and/or MQTT
Storage	Firebase or local server/storage
AI processing	CNN for images; Random Forest and XGBoost for sensor data
Development tools	Arduino IDE; Python/AI environment; CNN framework

4.1 Test Procedure

Greenhouse testing is intended to verify sensor acquisition, image capture, data transmission, AI inference, response time, false-positive behavior, battery operation, and overall reliability. After initial verification, field testing examines disease-detection capability, alert timeliness, robustness to weather and illumination variation, long-term monitoring, communication reliability, and farmer usability.

4.2 Disease-Classification Performance

The classification accuracy of 92% across 19 significant crop-disease categories and an 18–23% improvement over compared monocausal models.

Table II. Classification outcomes

Evaluation item	Value stated in project report
Number of labels/categories	19
Field classification accuracy	92%
Improvement over compared monocausal models	18–23%

The outcome is consistent with the design rationale that visual symptoms and environmental context may provide complementary information. However, the report does not include a confusion matrix, class-wise precision/recall, F1-score, confidence intervals, exact baseline identities, or a complete train/test protocol. Consequently, the

numerical claims cannot be independently reproduced from the supplied material.

4.3 Decision-Support and User-Engagement Outcomes

The farmer compliance with recommendations increased from 54% to 88% after adapting recommendations based on user responses. It also reports daily active usage of 94% for the proposed application compared with 31% for traditional agricultural applications. The source does not include participant counts, the definition of compliance or daily active use, observation duration, or participant-level records.

Table III. Decision-support and engagement outcomes

Parameter	Reported value
Initial recommendation compliance	54%
Improved recommendation compliance	88%
Proposed application daily active usage	94%
Traditional application daily active usage	31%

4.4 Interpretation

The paper results suggest that a crop-monitoring platform should be evaluated not only through classification accuracy but also through communication reliability, alert timeliness, and user acceptance. Multimodal analysis may offer a richer context than either image-only or sensor-only analysis. Likewise, low-bandwidth access and understandable alerts may influence whether farmers act on system outputs. These interpretations are qualitative in the absence of the detailed experimental records described above.

The proposed framework combines sensing, embedded communication, visual recognition, environmental classification, and decision support in a single architecture. Its central design principle is that crop-health assessment can benefit from more than one information source. Leaf images can capture visible symptoms, whereas environmental measurements describe conditions surrounding the plant. A fusion stage can use both sources to produce a contextual risk estimate.

The architecture also separates acquisition from computationally intensive AI inference. The ESP32 provides a practical interface to sensors and wireless communication, while CNN and tree-ensemble models can run on a local edge device or cloud server depending on the deployment configuration. This separation supports modular development and permits the AI models to be updated without redesigning the sensing layer. Local buffering is important because connectivity in agricultural fields may be intermittent.

Several implementation choices require careful validation. First, sensor readings should be calibrated and checked for drift, placement effects, and environmental exposure. Second, the image dataset should represent different crops, disease stages, illumination levels, camera distances, and field backgrounds. Third, the sensor model should be trained using measurements collected under conditions representative of deployment. Fourth, fusion weights and alert thresholds should be selected using validation data rather than adjusted on the final test set.

Image classification identifies visual patterns in an acquired image; it does not by itself establish that a disease will develop in the future. Environmental measurements may indicate conditions associated with risk, but a validated predictive claim requires time-stamped longitudinal observations and a clearly defined prediction horizon. Accordingly, the present framework is best described as crop-disease detection and risk assessment unless prospective temporal validation is completed.

6. CONCLUSION AND FUTURE WORK

This paper has presented an AIoT-based framework for smart crop-disease monitoring that integrates environmental sensing, camera-based image acquisition, ESP32-based embedded communication, AI classification, cloud/edge processing, and farmer-oriented decision support. Temperature, humidity, soil moisture, and light intensity provide environmental context, while leaf images provide visual information. CNN-based image classification and Random Forest/XGBoost sensor-data classification form complementary prediction paths. A weighted fusion stage combines their outputs to generate risk information and alerts.

The project report states a field classification accuracy of 92% across 19 labels/categories, an 18–23% improvement over compared monocausal models, recommendation compliance increasing from 54% to 88%, and application daily active usage of 94% compared with 31% for traditional applications. Future work can extend the dataset to additional crops, disease categories, and disease stages, with observations collected across different geographic and climatic conditions. Drone-based image acquisition may support monitoring of larger fields. More

efficient edge-AI models can reduce dependence on cloud connectivity and improve response time. Solar-powered operation and energy-efficient hardware can support long-term deployment in remote locations. Integration with agricultural advisory services may further connect model outputs with expert recommendations and regional crop-management information.

REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, vol. 7, article 1419, 2016.
[2] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," Applied Artificial Intelligence, vol. 31, no. 4, pp. 299–315, 2017.
[3] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.
[4] M. Zhang, Y. Meng, and Z. Gong, "An Intelligent IoT-Based System for Real-Time Vegetable Disease Detection Using Machine Learning," Sensors, vol. 19, no. 24, article 4471, 2019.
[5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
[6] Arduino, "ESP32-CAM Development Board," Arduino Documentation.
[7] TensorFlow, "TensorFlow Lite for Microcontrollers," TensorFlow Documentation.
[8] PlantVillage Dataset, "Plant Disease Classification," Kaggle.
[9] Google Firebase, "Firebase Realtime Database Documentation," Firebase Documentation.
[10] Food and Agriculture Organization, "Digital Agriculture and AI: Helping Farmers Make Better Decisions," FAO, 2021.
[11] G. Nisha and R. Ramya, "Smart Agriculture System Using IoT," International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 3, pp. 446–449, 2017.
[12] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Computational Intelligence and Neuroscience, vol. 2016, pp. 1–11, 2016.
[13] D. Patil and V. Kale, "Smart Soil Monitoring System using IoT and NPK Sensors," International Research Journal of Engineering and Technology, vol. 7, no. 6, pp. 1735–1739, 2020.
[14] R. Bhargava et al., "AI and IoT Based Smart Agriculture System," International Journal of Engineering Research & Technology, vol. 11, no. 2, pp. 120–125, 2022.
[15] J. G. Ahamed and M. J. Pathan, "A Review on IoT Based Applications in Smart Agriculture," International Journal of Advanced Research in Computer and Communication Engineering, vol. 8, no. 3, pp. 34–38, 2019.
[16] R. R. Rajalakshmi and S. Devi, "Real Time Smart Crop Protection System," International Journal of Innovative Research in Computer and Communication Engineering, vol. 5, no. 2, pp. 3021–3026, 2017.
[17] A. A. Deshmukh and S. D. Sawarkar, "Smart Farming System using IoT," International Journal of Computer Applications, vol. 179, no. 47, pp. 1–4, 2018.
[18] S. D. Mane and M. S. Ankalaki, "Review on Machine Learning Techniques for Crop Disease Prediction," International Journal of Innovative Science and Research Technology, vol. 6, no. 2, pp. 200–205, 2021.
[19] H. Dhruvika and A. Patel, "AI Based Crop Disease Prediction Using Image Processing Techniques," International Journal of Engineering and Advanced Technology, vol. 9, no. 3, pp. 514–519, 2020.
[20] M. Ghosh, A. Pal, and S. Bhattacharjee, "Crop Disease Prediction Using Machine Learning: A Review," International Journal of Engineering Research & Technology, vol. 9, no. 12, pp. 345–348, 2020.
[21] P. Kumar, M. Sharma, and R. Agarwal, "Sensor Based IoT Architecture for Smart Agriculture," International Journal of Computer Applications, vol. 180, no. 47, pp. 23–28, 2018.
[22] T. Jayaraman et al., "Applications of Artificial Intelligence and Machine Learning in Smart Agriculture," International Journal of Intelligent Systems and Applications in Engineering, vol. 9, no. 2, pp. 65–71, 2021.
[23] A. Singh and P. Yadav, "Real-Time Crop Health Monitoring and Protection System Using Deep Learning," International Journal of Scientific & Engineering Research, vol. 11, no. 6, pp. 678–682, 2020.
[24] A. Jain and N. Garg, "Smart Farming Using IoT and Machine Learning for Crop Health Monitoring," International Journal of Computer Sciences and Engineering, vol. 8, no. 5, pp. 77–83, 2020.
[25] S. V. Bhadoria and R. K. Tripathi, "Emerging Technologies in Agriculture: Role of AI and IoT," International Journal of Recent Technology and Engineering, vol. 8, no. 5, pp. 234–239, 2020.