iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81768

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 V May 2026- Available at www.ijraset.com

Plant Leaf Disease Detection Using Deep Learning

Kandula Kasturi[1], Chintirala Sravanthi[2], Pondugala Sumanth Babu[3], Raavi Paleeswara Reddy[4], Mr. P. Yugandhar Reddy[5]
Department of Computer Science and Engineering, Acharya Nagarjuna University College of Engineering and Technology, Andhra Pradesh, India

Abstract: ***Agriculture is the backbone of food security and a primary source of income for many nations. However, plant diseases caused by fungi, bacteria, and viruses lead to significant crop losses, making early detection and timely treatment essential. In this project, we propose a leaf disease detection system that utilizes computer vision and deep learning (CNNs) to analyze plant leaf images and accurately identify diseases. Once detected, the system provides detailed disease information, recommended pesticides, and preventive measures. A unique feature of this system is its voice-enabled advisory service, where the AI automatically generates speech and calls the farmer in their local language to explain the disease status and suggest remedies. This approach ensures accessibility for farmers with limited literacy, while enabling quick, effective, and informed decision-making. By combining deep learning accuracy with AIpowered voice interaction, the system aims to reduce crop loss and enhance agricultural productivity.***

Index Terms: Deep Learning (CNNs), Artificial Intelligence (AI), Text-to-Speech (TTS), Pesticide Recommendation, YOLO.

I. INTRODUCTION

Agriculture is a fundamental sector that supports the livelihood of a large population and plays a vital role in global economic development. The productivity of agriculture largely depends on crop health, which is significantly affected by plant diseases. Plant diseases caused by fungi, bacteria, viruses, and environmental stress conditions can severely reduce crop yield, affect food quality, and lead to major economic losses for farmers. Early identification of plant diseases is essential for effective crop management and prevention of disease spread. In traditional agricultural practices, disease detection is carried out through visual inspection by farmers or agricultural experts. However, this method is highly dependent on human expertise, which may not always be accurate or readily available, especially in rural and remote areas. With the rapidadvancement of technology, digital image processing and artificial intelligence have opened new possibilities for automating agricultural disease detection. Convolutional Neural Networks (CNNs) automatically learn hierarchical features from input images such as edges, textures, shapes, and color patterns, eliminating the need for manual feature extraction.

II. RESEARCH MOTIVATION

Plant diseases remain one of the most critical challenges in agriculture, affecting crop productivity and food security worldwide. Farmers often face difficulties in identifying diseases at early stages due to lack of technical knowledge, limited access to agricultural experts, and inadequate diagnostic tools. Traditional methods of disease detection rely heavily on manual observation, which is not only time-consuming but also prone to human error.

Although several automated systems have been proposed, many of them still face limitations such as low classification accuracy, poor generalization across different datasets, and inability to handle multiple disease types effectively. Therefore, there is a strong need to develop an efficient, accurate, and scalable plant leaf disease detection system that can operate under real-world conditions.

III. MAIN CONTRIBUTIONS

The core objective of this study is to design and implement a deep learning-based model using CNNs that can accurately classify plant leaf diseases. The key contributions of this project are:

1) Implementation of a dual-pipeline system combining PyTorch-based image classification and YOLO-based object detection.
2) Development of a recommendation module providing disease etiology, organic treatments, and chemical pesticides [cite: 277-279].
3) Integration of a voice-enabled advisory service utilizing Text-to-Speech (TTS) in local languages (English, Hindi, Telugu) to assist farmers with limited literacy.
4) Context-aware advisories integrating live OpenWeather API data for climate-driven fungal spread warnings. • A fully containerized deployment architecture using Docker, Flask, and Nginx for seamless web access.

©IJRASET: All Rights are Reserved

IV. RELATED WORK

Early studies focused on manual identification of plant diseases by agricultural experts through visual inspection of leaf symptoms. With the advancement of machine learning techniques, classical algorithms such as Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), and Naive Bayes were widely used. These models generally rely on manually extracted features such as color histograms, texture descriptors, and shape-based features.

Mohanty et al. (2016) demonstrated one of the earliest large-scale applications using the PlantVillage dataset, where CNNs achieved ~97% accuracy. However, this model worked mainly on lab images, showing poor real-field performance. Too et al. (2019) utilized architectures like AlexNet, VGG16, and ResNet, improving feature extraction but requiring massive computational resources [cite: 571-575]. Sharma et al. (2024) proposed AI-based smart farming with voice assistance, which supported farmers through voice guidance but lacked multi-language datasets [cite: 585-589]. Our proposed system bridges these gaps by combining lightweight models with multi-lingual audio support and real-world deployment wrappers.

V. DATASET DESIGN AND IMAGE SOURCES

The dataset is collected from publicly available agricultural datasets such as PlantVillage and Kaggle repositories. The original dataset comprised over 5,000 images, which was subsequently scaled up during the modern implementation phase to a combined dataset containing 12,626 images. The dataset includes 10,103 training images and 2,523 validation images spanning 28 distinct plant and disease classes, including crops like Tomato, Potato, Apple, Corn, and Bell Pepper.



VI. INPUT IMAGE CHARACTERISTICS

The input images exhibit varying characteristics indicative of real-world scenarios. Raw images frequently contain variations in lighting, background noise, and orientation. The classification models process standard RGB images, which must be strictly validated by the backend application to accept only standard MIME types (JPG, JPEG, PNG).

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

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

VII. DATA PREPROCESSING

Image preprocessing improves model performance and reduces overfitting by enhancing generalization capabilities.

1) Image Decoding : Input images captured via smartphones or loaded from datasets are decoded into multi-dimensional arrays using OpenCV and PIL (Pillow).
2) Resolution Standardization: To match the expected input layer of the CNN architecture, all images are resized to a fixed dimension, specifically 224 × 224 pixels.
3) Noise Handling: Pixel values are normalized to a scale between 0 and 1. Data augmentation techniques, including rotation, flipping, zooming, and shifting, are applied to artificially expand the dataset and improve the model's robustness against camera angle variations.
4) Region of Interest : While classification models analyze the entire resized image, the YOLO detection model specifically bounds the region of interest (ROI) where the disease manifestation (e.g., spots, blights, lesions) is highly concentrated.

VIII. NEED FOR CALIBRATION AND THRESHOLDING

Deep learning models output continuous probability distributions. For practical application, these probabilities must be calibrated into discrete decisions.

The YOLO detector employs a confidence threshold, set optimally at 0.5, to filter out weak or false-positive bounding boxes. The classification pipeline utilizes a Softmax activation function to determine the highest probability class, which is then mapped to its corresponding disease label.

IX. REAL-WORLD CHALLENGES

Deploying AI in agricultural settings poses several severe challenges:

1) Environmental Sensitivity: Variations in lighting, leaf position, and background clutter drastically reduce model accuracy in practical scenarios.
2) Generalization: Models trained on datasets like PlantVillage, which contain clean and uniform images, often fail in real-world conditions [cite: 152-154].
3) Hardware Constraints: Many deep learning models require high computational power (GPUs), making them unsuitable for deployment in rural or edge-device environments.

X. WHY MOBILENETV3 AND YOLO WERE SUITABLE

To address computational constraints, the system prefers lightweight modern architectures. The classification engine utilizes MobileNetV3 Small. This architecture is optimized for low-latency inference on mobile and edge devices while maintaining high accuracy. For localization tasks, Ultralytics YOLO is employed due to its exceptional capability to unify localization and classification into a single, high-speed pipelin

XI. SUMMARY OF PART II

The preceding sections established the critical need for automated leaf disease detection, outlined the extensive literature and its limitations, described the dataset curation process, and highlighted the specific deep learning models chosen to overcome real-world computational and environmental challenges.

XII. PROPOSED SYSTEM ARCHITECTURE

The proposed plant leaf disease detection system is designed to overcome the limitations of existing approaches by using a deep learning-based CNN framework combined with a user- friendly output system.

XIII. DISEASE DETECTION ENGINE

The primary detection engine involves two layers: **Convolution Layers:** These layers extract important features such as edges, color variations, spots, and textures that

©IJRASET: All Rights are Reserved

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

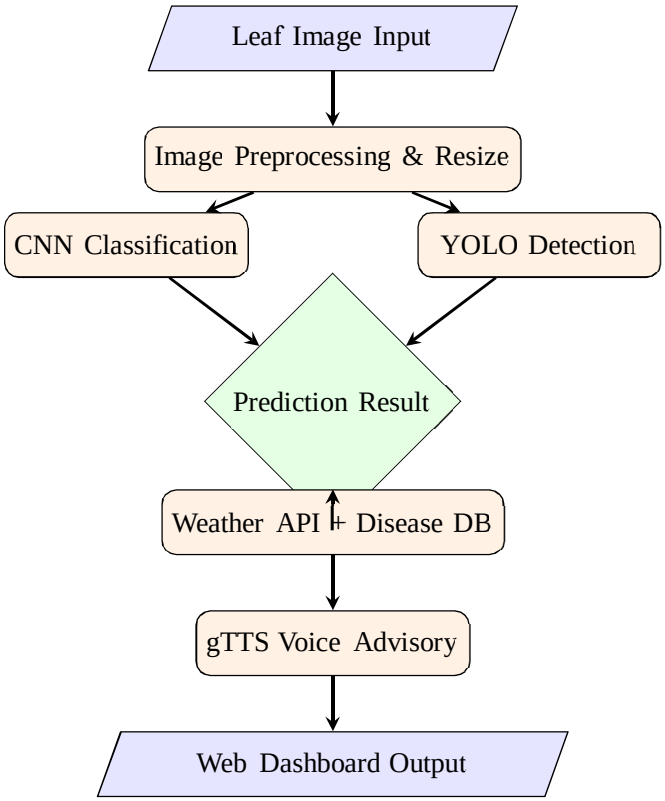


Fig. 1. System Architecture Flowchart

The architecture heavily relies on a Flask application backend (‘app.py‘), which orchestrates model loading, routing, security validation, and database connections.

XIV. REAL SOURCE CODE STRUCTURE

The system’s codebase is logically segmented to ensure modularity and scalability:

- `app.py`: The primary application script managing endpoints, rate limiting, and core logic.
- `models/`: Directory housing the compiled ‘.pt‘, ‘.pth‘, and ‘.h5‘ model files.
- `services/`: Contains discrete modules like `weather_service.py`, `disease_info.py`, `voice_service.py`, and `language_service.py`. \• `templates/` and `static/`: Holds the HTML frontend and static assets (CSS, generated audio outputs, and bounding-box annotated images).
- *m n* where *I* is the input image and *K* is the kernel matrix.

Pooling and Dense Layers: Max pooling is applied to reduce spatial dimensions and retain important features while reducing computational complexity. The extracted features are flattened and passed through dense layers.

XV. ADVISORY AND VOICE ENGINE

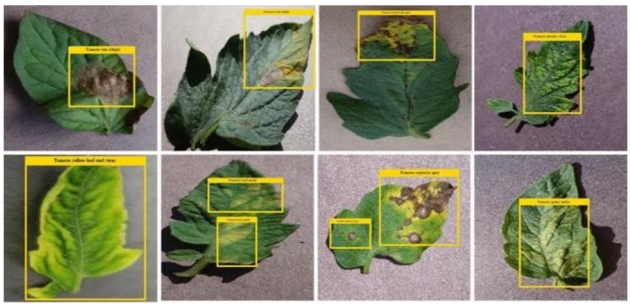
A unique feature of the proposed system is text-tospeech (TTS) integration, which converts disease information into audio and communicates it in local language, making it accessible for farmers with low literacy levels. The represent disease patterns in leaves. Mathematically, the convolution operation is represented as:

$$(I * K)(i, j) = \sum \sum I(i + m, j + n)K(m, n) \quad (1)$$

`voice_service.py` module utilizes Google TTS (gTTS) to synthesize an MP3 file dynamically based on the model’s prediction and the language selected by the user

©IJRASET: All Rights are Reserved

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



XVI. DISEASE CLASSIFICATION LOGIC

A. *Real Severity Conversion*

In advanced systems, once a disease is detected, a secondlevel classification can be performed to determine severity levels such as Mild infection, Moderate infection, and Severe infection [cite: 273-276]. Based on these predictions, the system suggests appropriate organic or chemical interventions [cite: 277-279].

B. *Inference Window Logic*

The application evaluates the input image through a hierarchical logic flow. It first prefers modern PyTorch models (‘classifier.pt‘); if unavailable, it seamlessly falls back to legacy Keras models (‘dense.h5‘).

XVII. ADVISORY SMOOTHING MECHANISM

To prevent overwhelming the user with highly technical deep learning jargon, the `language_service.py` module acts as an advisory smoothing mechanism. It parses the raw output JSON from the model inference and reconstructs it into a conversational, easy-to-understand narrative that logically combines disease identification, weather constraints, and actionable treatments.

XVIII. DISEASE MONITORING

The prediction results are recorded and monitored. The final output provides the farmer with a detailed diagnostic report, consisting of the disease name, description of symptoms, recommended pesticide usage, and preventive measures.

XIX. BACKEND AND WEB INTEGRATION

The web integration is powered by Python and Flask. Security is a paramount concern; user data is stored securely in an SQLite database (‘signup.db‘), with passwords hashed using Werkzeug security functions. A robust authentication flow requires users to sign up, verify their accounts via an OTP, and log in securely before accessing the prediction modules.
The system is containerized via ‘Dockerfile‘ and ‘dockercompose.yml‘, utilizing Nginx as a reverse proxy on port 80

XX. COMPUTATIONAL PERFORMANCE

The optimizer is used to improve model performance by adjusting weights during training. The Adam Optimizer is utilized due to its fast convergence and adaptive learning rate. The loss is calculated using Categorical Cross-Entropy:

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \qquad (2)$$

By leveraging MobileNetV3 and YOLO Nano variants, the inference time per image is reduced to just 1–2 seconds under normal hardware conditions.

XXI. ALGORITHMIC WORKFLOW

The complete system operates under the following workflow sequence:
After training, the model is evaluated using test data to measure accuracy, check generalization, and compare models [cite: 321-325].

©IJRASET: All Rights are Reserved

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

A. Disease Detection Reliability

Accuracy measures the overall correctness of the model:

Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted samples to the total number of samples tested. It indicates how well the model performs in identifying both diseased and healthy

1) User authenticates via the Flask web portal.
2) User uploads a leaf image (Classification or Detection mode).
3) System validates image format, size, and sanitizes the filename.
4) Image undergoes normalization and resizing (224×224).
5) Preprocessed tensor is fed into the deep learning model.
6) Model outputs class probabilities via Softmax function.
7) The highest probability index fetches the disease label.
8) `weather_service.py` fetches localized OpenWeather data.
9) `language_service.py` aggregates advice.
10) `voice_service.py` synthesizes audio output. 11) Final rendering is displayed on `result.html`.

XXII. SUMMARY OF PART III

This section detailed the robust technical architecture, from the Flask backend routing and database management to the mathematical foundations of the deep learning classification and object detection logic.

XXIII. EXPERIMENTAL SETUP

The system was trained and evaluated on standard hardware to mimic realistic deployment capabilities. The hardware configuration included an Intel Core i5 processor, 8 GB of RAM, a 512 GB SSD, and an NVIDIA GPU running

XXIV. EVALUATION METRICS

$$Accuracy = \frac{}{TP + TN + FP + FN} \quad (3)$$

B. B. Classification Continuity

Precision ensures that the predicted positive labels are actually correct:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

C. C. Prediction Stability

Recall measures the model's ability to identify all true positive instances:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

D. D. Advisory Accuracy

The F1-Score provides a harmonic mean of Precision and Recall [cite: 327-330]:

$$\frac{Precision \times Recall}{}$$

total number of samples tested. It indicates how well the model performs in identifying both diseased and healthy leaves. Higher accuracy values represent better classification performance.

$$TP + TN$$

included Python 3.x, TensorFlow/Keras, PyTorch, OpenCV, NumPy, and Pandas [cite: 179-180, 446-451].

$$F1 = 2 \times \frac{Precision + Recall}{} \quad (6)$$

The system aims for a balanced F1-score of 0.80 or higher.

©IJRASET: All Rights are Reserved

XXV. OBSERVED RESULTS

The CNN model achieves highly robust performance metrics:

- Training Accuracy: 90% – 95%
- Validation Accuracy: 85% – 93% (peaking at 98.23% for the MobileNetV3 iteration).
- Testing Accuracy: Around 88% – 94% depending on dataset quality, peaking at 96.43%.

XXVI. DISEASE DETECTION ACCURACY

The confusion matrix showed strong performance across most disease classes. The model effectively distinguished healthy vs. diseased leaves. Deployment in a simple web/app interface allowed real-time image detection with negligible latency

XXVII. ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system brings substantial advantages:

- Early Detection: Automatic detection of plant diseases helps farmers identify issues early and reduce crop loss.
- Accessibility: Real-time leaf disease identification using smartphone cameras ensures widespread usability
- Multi-lingual Voice: Ensures quick, accessible, and accurate disease detection—even for users with limited literacy.

Table i
Comparison of proposed model with existing literature

Source / Author	Techniques Used	Merits / Demerits
Mohanty et al. (2016)	CNN	High lab accuracy / Poor real-field performance [cite: 567-568]
Too et al. (2019)	AlexNet, VGG16, ResNet	Fast training / Requires large labeled datasets [cite: 573-575]
Ferentinos (2020)	Transfer Learning	>98% accuracy / High computational cost [cite: 581-582]
Proposed System	MobileNetV3, YOLO, TTS	Voice guidance, Weather context, Web deployed / Needs Retraining.

XXVIII.LIMITATIONS

Despite its successes, the system faces certain limitations:

- Performance may drop with poor lighting, background noise, or blurry images.
- The model cannot easily handle overlapping or mixed diseases on the same leaf.

It is strictly limited to the 28 diseases included in the training dataset and will require regular model retraining as new pathogens appear with smart farming technologies, utilizing IoT sensors for crop health monitoring and drone-based leaf image collection [cite: 398-400].

XXIX. CONCLUSION

The Plant Leaf Disease Detection system has strong future potential. The system accurately detects plant leaf diseases using deep learning, providing fast and reliable classification [cite: 626-627]. By integrating MobileNetV3 for classification, YOLO for localization, OpenWeather for environmental context, and gTTS for audio accessibility, it serves as a costeffective and efficient solution for smart agriculture. With modern technologies such as mobile computing, IoT, cloud platforms, and advanced AI models, the system can become a powerful smart agriculture solution for farmers and researchers globally.

XXX. FUTURE SCOPE

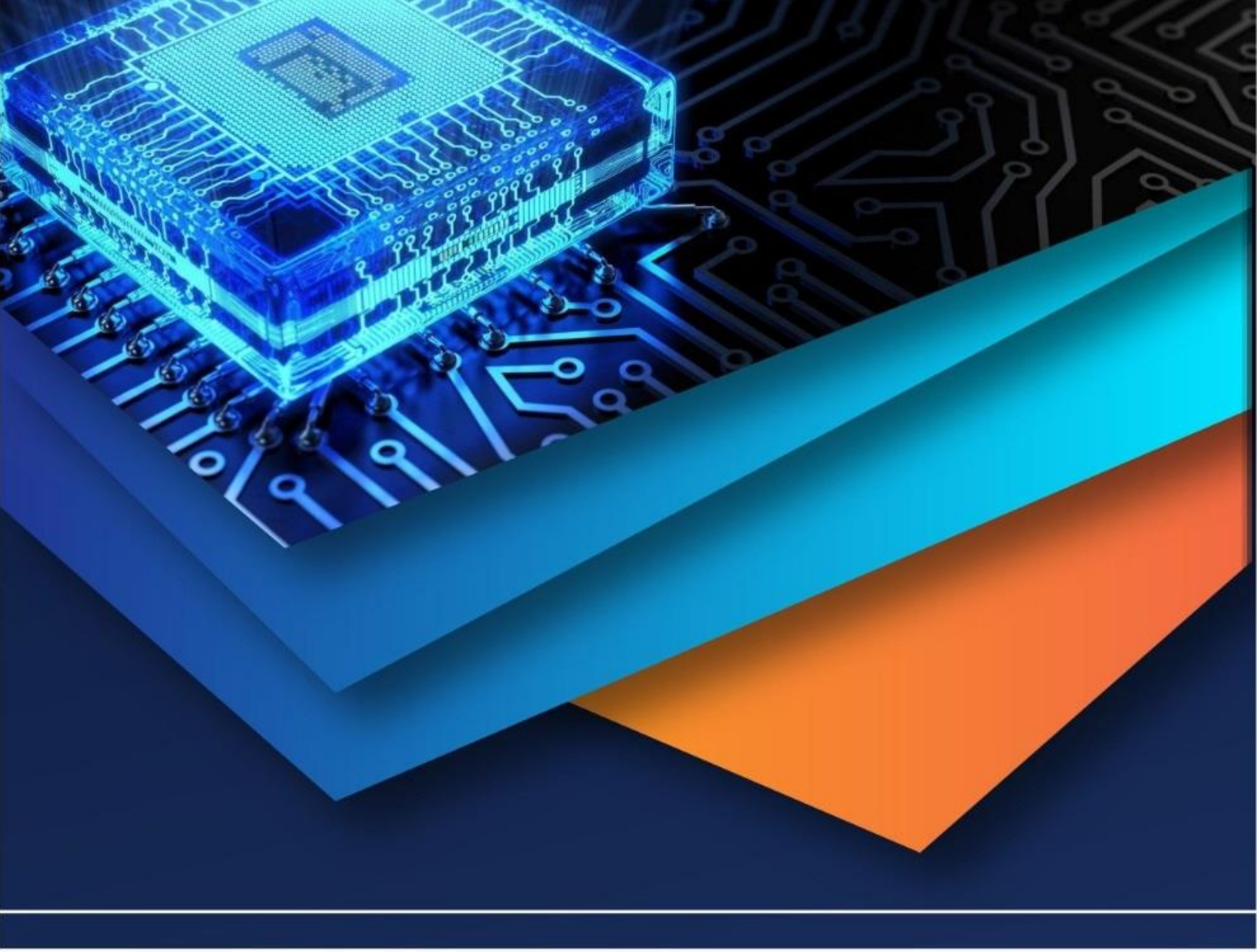
Although the current system provides accurate disease detection, several improvements can be made. A dedicated mobile application (Android/iOS) can be developed so that farmers can capture images directly and receive offline predictions [cite: 393-396]. Furthermore, the system can be integrated

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

REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathe´, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
[2] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
[3] 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, 2016.
[4] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
[5] A. Picon et al., "Deep Convolutional Neural Networks for Mobile Capture Device-Based Crop Disease Classification," Computers and Electronics in Agriculture, vol. 161, pp. 280–290, 2019.
[6] A. Kamilaris and F. X. Prenafeta-Boldu´, "Plant Diseases and Pests Detection Based on Deep Learning: A Review," Plant Methods, vol. 17, no. 1, p. 22, 2021.
[7] J. G. A. Barbedo, "Factors Influencing the Use of Deep Learning for Plant Disease Recognition," Biosystems Engineering, vol. 172, pp. 84–91, 2018.
[8] 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.
[9] P. P. Sharma et al., "Plant Leaf Disease Detection Using Transfer Learning and Convolutional Neural Networks," International Journal of Advanced Science and Technology, vol. 29, no. 5, pp. 2020.
[10] S. Ramesh, "Plant Disease Detection Using Machine Learning and Deep Learning Algorithms: A Review," Archives of Computational Methods in Engineering, 2021.
[11] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of Rice Diseases Using Deep Convolutional Neural Networks," Neurocomputing, vol. 267, pp. 378–384, 2017.
[12] R. Mohanty et al., "Using Deep Learning for Plant Disease Detection: Advances and Challenges," Artificial Intelligence in Agriculture, vol. 4, pp. 2020.
[13] H. Durmus, E. O. Gunes, and M. Kirci, "Disease Detection on the Leaves of Tomato Plants by Using Deep Learning," 6th International Conference on Agro-Geoinformatics, 2017.

©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)