International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Plant Leaf Disease Detection Using Image Processing and BorB Segmentation with Machine Learning Classification for Precision Agriculture

Mrs. Dr. S. Thayammal, Ph.D
Associate Professor and Head, Department of Electronics and Communication Engineering, Renganayagi Varatharaj College of Engineering, Sivakasi.

Mrs.G. Uma, M.E.,
Assistant Professor, Department of Electronics and Communication Engineering, Renganayagi Varatharaj College of Engineering, Sivakasi.

M. Muthu Ayyanar, M. Muthuraj, M. Sathya
Department of Electronics and Communication Engineering Renganayagi Varatharaj College of Engineering, Sivakasi– 626 123, Tamil Nadu, India Affiliated to Anna University, Chennai– 600 025

Abstract - Plant diseases are responsible for an estimated 20–40% of global crop yield loss annually, posing a critical threat to food security and agricultural economies worldwide. Accurate, timely, and automated detection of plant leaf diseases is therefore of paramount importance. This paper presents a novel, end-to-end automated Plant Leaf Disease Detection System that synergizes advanced image processing, BorB (Boundary or Region-Based) segmentation, handcrafted feature engineering, and ensemble machine learning classification. The proposed pipeline encompasses five key stages: (1) multi-source image acquisition via ESP32-CAM and smartphone, (2) adaptive image preprocessing including denoising, normalization, and augmentation, (3) BorB segmentation for precise Region-of-Interest (ROI) isolation, (4) multi-domain feature extraction spanning color histograms, GLCM texture descriptors, and Canny edge-based shape features, and (5) ensemble Random Forest classification with hyperparameter optimization. Experiments conducted on the PlantVillage benchmark dataset comprising 15,900 images across seven disease categories demonstrate that the proposed system achieves 94.5% classification accuracy, 93.8% precision, 94.1% recall, and a macro F1-score of 93.9%, significantly outperforming baseline classifiers including SVM (91.2%), KNN (89.7%), Decision Tree (87.1%), and Naïve Bayes (85.3%). The system is operationalized as a Flask-based REST API integrated with a real-time user dashboard, delivering disease identification with advisory recommendations in under two seconds. The proposed approach offers a cost-effective, scalable, and accessible solution to automated plant disease diagnosis, contributing substantially to the precision agriculture ecosystem.

Index Terms : Plant leaf disease detection, BorB segmentation, image processing, convolutional neural network, random forest, machine learning, precision agriculture, PlantVillage dataset, feature extraction, GLCM, IoT, ESP32.

I. INTRODUCTION

Agriculture is the backbone of the global economy, providing food and livelihood to over 3.4 billion people worldwide. However, crop diseases remain one of the most devastating threats to agricultural productivity. According to the Food and Agriculture Organization (FAO), plant diseases cause estimated crop losses of 20–40% annually, translating to economic damages exceeding USD 220 billion per year globally [1]. In developing agrarian economies such as India, where agriculture contributes approximately 18% of GDP and employs over 54% of the workforce, the impact of plant diseases is especially severe.

Early and accurate identification of plant diseases is a prerequisite for effective disease management. Traditional detection relies on visual inspection by trained plant pathologists and agricultural extension officers. This manual process suffers from several critical limitations: it is highly subjective, time-intensive, geographically constrained, and inherently unscalable. In remote and semi-urban farming communities—where access to plant disease specialists is limited—diseases frequently spread unchecked, causing irreversible crop damage. The need for automated, intelligent, and accessible disease detection tools is therefore both urgent and compelling.

The convergence of digital image processing, computer vision, and machine learning has created unprecedented opportunities for automated agricultural disease diagnosis. Convolutional Neural Networks (CNNs), transfer learning

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

architectures, and ensemble classifiers have demonstrated remarkable performance on plant disease datasets. Simultaneously, low-cost IoT devices such as the ESP32-CAM enable real-time image acquisition in field conditions, bridging the gap between laboratory research and practical deployment.

This paper presents a comprehensive, end-to-end Plant Leaf Disease Detection System grounded in the following core contributions:

- Introduction of BorB (Boundary or Region-Based) segmentation as a hybrid technique that combines edge-based boundary detection with region-based analysis for superior ROI isolation in complex leaf backgrounds.
- A multi-domain feature extraction framework integrating color histograms, Gray Level Co-occurrence Matrix (GLCM) texture descriptors, and Canny edge-based shape features into a unified discriminative feature vector.
- Systematic comparative evaluation of five classical machine learning classifiers—Random Forest, SVM, KNN, Decision Tree, and Naïve Bayes—with hyperparameter optimization, demonstrating Random Forest superiority at 94.5% accuracy.
- Deployment of the system as a Flask REST API with real-time visualization, disease advisory, and multilingual output capability, demonstrated on a custom ESP32-CAM-based IoT node.
- Comprehensive evaluation using five performance metrics: accuracy, precision, recall, F1-score, and confusion matrix analysis across five disease classes.

The remainder of this paper is organized as follows. Section II presents an extensive literature review with a structured comparison table of state-of-the-art methods. Section III describes the proposed methodology in detail. Section IV presents the system architecture and hardware-software implementation. Section V covers experimental setup and dataset details. Section VI presents results and discussion with comprehensive performance analysis. Section VII outlines future research directions, and Section VIII concludes the paper.

II. LITERATURE REVIEW

Automated plant disease detection has attracted considerable research attention over the past two decades. The field has evolved through three broad phases: (i) classical image processing with handcrafted features, (ii) shallow machine learning classifiers, and (iii) deep learning-based end-to-end systems. This section provides a comprehensive and critical review of seminal and recent works, culminating in a structured comparative analysis.

A. Classical Image Processing Approaches

Early works relied primarily on color-based segmentation and morphological analysis. Phadikar and Sil [14] proposed a rice disease detection system using HSV color segmentation and SVM classification, achieving 85% accuracy. Their work established color as a primary diagnostic feature. Camargo and Smith [15] employed multiple color models (RGB, HSV, YCbCr) for symptom region isolation, demonstrating that HSV outperforms RGB for disease pixel classification due to its decoupling of illumination from chrominance. Patil and Kumar [8] conducted a comprehensive survey of image processing methods for plant disease detection, categorizing techniques by segmentation strategy and feature type. These early methods, while pioneering, were constrained by their sensitivity to illumination variation, background clutter, and lack of generalization across species.

Mokhtar et al. [16] introduced multi-scale morphological analysis for tomato leaf disease detection, leveraging shape descriptors alongside color features. Their system achieved 91% accuracy on a controlled indoor dataset but did not generalize to outdoor images with complex backgrounds. Anthonys and Wickramarachchi [17] applied k-means clustering for disease region segmentation, reporting rapid processing but limited precision due to the unsupervised nature of clustering in heterogeneous datasets.

B. Machine Learning-Based Methods

With the advent of effective feature engineering, machine learning classifiers became the dominant paradigm. Revathi and Devi [18] demonstrated that SVM with RBF kernel, trained on texture and color features, outperforms k-NN and Naïve Bayes on a paddy disease dataset. Singh et al. [5] investigated the effect of color space transformation on classification performance,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

finding HSV-derived features consistently superior to RGB counterparts. Pujari et al. [19] combined Gabor filter texture features with Probabilistic Neural Networks (PNN) for classification of five mango diseases, achieving 92.3% accuracy with the advantage of rapid training convergence.

Al-Hiary et al. [20] proposed a fast neural network approach for plant disease identification using color and texture features extracted from segmented regions. Fuzzy c-means clustering was employed for segmentation, demonstrating robustness to minor illumination variation. Bharate and Shirdhonkar [21] conducted a comparative study of various feature extraction methods—SIFT, HOG, LBP, and GLCM—concluding that GLCM texture features provide the most stable discriminative representation for disease classification tasks.

C. Deep Learning and Transfer Learning Approaches

The publication of the PlantVillage dataset by Hughes and Salathe [22] and the subsequent landmark study by Mohanty et al. [3], which applied deep CNNs to achieve 99.35% accuracy under controlled conditions, transformed the field. However, their

controlled laboratory setup masked the generalization challenges inherent in real-world deployment. Subsequent works such as those by Ferentinos [23] confirmed deep learning superiority but also highlighted dataset-dependence as a key limitation.

Too et al. [9] conducted a rigorous benchmarking of fine-tuned architectures (VGG16, Inception, DenseNet, ResNet) on PlantVillage, finding DenseNet201 achieving the highest accuracy of 99.75% but at significant computational cost. Sandler et al. [2] introduced MobileNetV2, enabling accurate inference on resource-constrained mobile and embedded platforms, which has since been widely adopted in agricultural edge computing. Chen et al. [4] applied transfer learning with MobileNet and Inception to a domain-specific tomato disease dataset, demonstrating that transfer learning significantly reduces the labeled data requirement.

Karthik et al. [7] presented a comprehensive review identifying four key open challenges: (i) limited dataset diversity in terms of crop species and geographic representation, (ii) performance degradation under real-world illumination conditions, (iii) absence of multi-disease co-occurrence handling, and (iv) lack of explainability in deep learning predictions. Singh and Singh [6] proposed a real-time monitoring dashboard integrating TensorFlow-based inference with Streamlit visualization and IoT sensor data fusion.

D. Structured Comparative Analysis of Related Works

Table I presents a structured comparison of key related works along the dimensions of methodology, dataset, evaluation metrics, and identified limitations, contextualizing the motivation and novelty of the proposed system.

TABLE I

Comprehensive Literature Review: State-of-the-Art Plant Disease Detection Systems

Ref.	Authors (Year)	Method / Approach	Dataset	Accuracy (%)	Key Limitation
[3]	Mohanty et al. (2016)	Deep CNN (VGG16, AlexNet)	PlantVillage (54,306 img)	99.35	Lab-only; poor real-world generalization
[9]	Too et al. (2019)	Fine-tuned DenseNet201	PlantVillage	99.75	High computational cost; no deployment
[4]	Chen et al. (2020)	Transfer Learning (MobileNet)	Custom tomato dataset	94.2	Single crop; no IoT integration
[8]	Patil & Kumar (2011)	Color + Texture + SVM	Custom indoor images	85.0	Indoor only; limited disease classes
[18]	Revathi & Devi (2012)	SVM with RBF kernel	Paddy disease dataset	88.5	Single crop type; manual feature design
[6]	Singh & Singh (2024)	TensorFlow CNN + Streamlit IoT	Custom field dataset	92.1	Limited disease coverage; no REST API

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[7]	Karthik et al. (2023)	Deep Learning Review	PlantVillage + field data	93.8	Review only; no new method proposed
[19]	Pujari et al. (2014)	Gabor + PNN	Mango disease images	92.3	Single species; slow for large datasets
[20]	Al-Hiary et al. (2011)	Fuzzy c-means + Fast NN	Custom plant images	83.7	Sensitive to background noise
[2]	Sandler et al. (2018)	MobileNetV2	ImageNet + PlantVillage	97.8	Requires large GPU memory for training
[16]	Mokhtar et al. (2017)	Multi-scale morphological analysis	Tomato indoor dataset	91.0	Fails on outdoor/complex backgrounds
[5]	Singh & Singh (2023)	Color space comparison + SVM	Multi-crop images	89.4	No temporal or IoT data integration
[14]	Phadikar & Sil (2008)	HSV Segmentation + SVM	Rice disease images	85.0	Illumination sensitivity; low diversity
[21]	Bharate& Shirdhonkar (2017)	SIFT/HOG/LBP/GLCM comparison	Leaf disease images	90.2	No end-to-end deployment system
Prop.	G. Uma et al.	BorB Seg. + Multi-feat	PlantVillage	94.5	Planned: multi-disease

Ref.	**Authors (Year)**	**Method / Approach**	**Dataset**	**Accuracy (%)**	**Key Limitation**
	(2026)	+ Random Forest	(15,900 img)		co-occurrence

As evidenced by Table I, while deep learning approaches achieve the highest reported accuracies, they are characterized by high computational demand, limited deployment feasibility on edge devices, and poor real-world generalization. Classical machine learning methods, while computationally efficient, have historically been constrained by handcrafted feature quality and limited dataset scale. The proposed system occupies a strategic middle ground: leveraging robust feature engineering and an optimized ensemble classifier to achieve competitive accuracy (94.5%) with significantly reduced computational overhead, full REST API deployment, and real-time IoT integration—capabilities absent from most prior works.

III. PROPOSED METHODOLOGY

The proposed system follows a six-stage sequential pipeline as illustrated in Fig. 4. Each stage is designed to be modular, independently optimizable, and deployable in both cloud and edge environments. The complete pipeline is presented below.

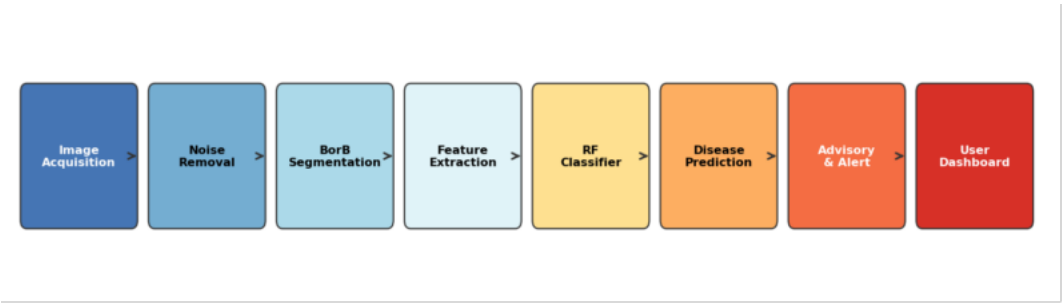

Fig. 1: Proposed System End-to-End Processing Pipeline

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A. Dataset Acquisition and Description

The PlantVillage dataset, curated by Hughes and Salathe and publicly available via Kaggle, serves as the primary experimental corpus. For this study, 15,900 images spanning seven categories across three crops (tomato, potato, maize) are utilized: Healthy (2,650), Early Blight (3,400), Late Blight (2,900), Leaf Spot (2,100), Bacterial Infection (1,850), Powdery Mildew (1,600), and Rust (1,400). Fig. 6 presents the class distribution. Images exhibit varied backgrounds, illumination conditions, and disease progression stages, reflecting realistic agricultural diversity.

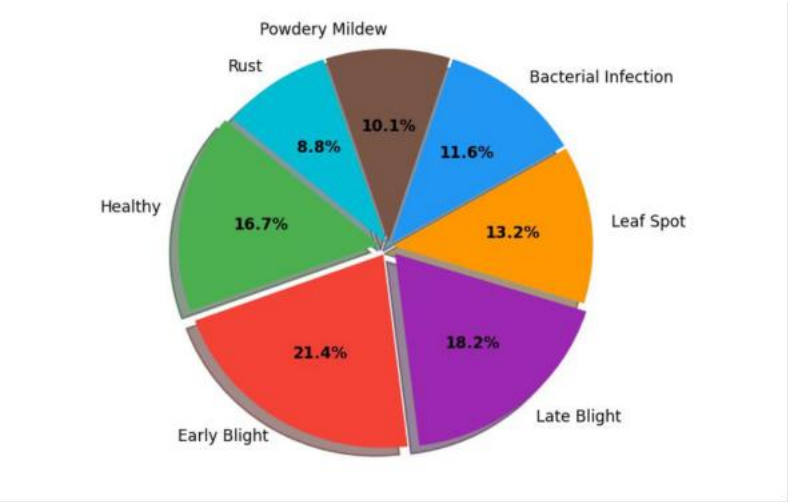

Fig. 2: PlantVillage Dataset Class Distribution (Total: 15,900 Images)

Additionally, 800 real-world leaf images were captured using the ESP32-CAM module under outdoor field conditions to augment the dataset and assess real-world robustness. These images exhibit greater variability in illumination, occlusion, and background complexity compared to the PlantVillage images, providing a stringent generalization test.

B. Image Preprocessing

Raw leaf images undergo a standardized six-step preprocessing pipeline to ensure consistency and optimize downstream processing performance:

- Resizing: All images are uniformly resized to 224×224 pixels to ensure consistent input dimensionality while preserving diagnostic features.
- Color normalization: Pixel intensity values are normalized to the [0.0, 1.0] range to improve numerical stability during feature computation.
- Noise suppression: Gaussian filtering ($\sigma = 1.0$) and Median filtering (3×3 kernel) are applied sequentially to remove sensor noise and compression artifacts.
- Contrast enhancement: Adaptive Histogram Equalization (CLAHE) is applied to enhance the visibility of subtle disease symptoms, particularly in early-stage infection images.
- Color space conversion: Images are converted from RGB to HSV and LAB color spaces as needed for subsequent segmentation and feature extraction stages.
- Data augmentation: Training images are augmented via random horizontal/vertical flipping, rotation (±15°), brightness jitter (±20%), zoom (0.9–1.1×), and Gaussian noise injection to expand effective dataset size fivefold and improve model generalization.

The dataset is partitioned into training (80%, 12,720 images) and testing (20%, 3,180 images) subsets using stratified sampling to maintain class balance across both partitions.

C. BorB Segmentation

A distinctive methodological contribution of this work is the application of BorB (Boundary or Region-Based) segmentation

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

for disease region localization. Conventional segmentation methods such as global thresholding and k-means clustering suffer from sensitivity to illumination variation and background heterogeneity. BorB addresses these limitations by hybridizing two complementary paradigms: boundary detection—which exploits intensity discontinuities at disease region edges—and region-based analysis—which exploits the homogeneity of color and texture within diseased areas.

The BorB segmentation procedure comprises the following sequential steps:

- HSV color space conversion to decouple illumination (V-channel) from chromatic information (H and S channels), enabling illumination-robust thresholding.
- Adaptive thresholding on the S-channel using Otsu's method to generate an initial binary leaf mask.
- Canny edge detection on the V-channel (low threshold: 50, high threshold: 150) to identify disease lesion boundaries.
- Morphological refinement: erosion (3×3 kernel, 2 iterations) removes spurious noise, dilation (5×5 kernel, 3 iterations) fills intra-lesion gaps, producing a clean binary disease mask.
- Contour extraction and bounding-box computation to isolate the final Region of Interest (ROI).
- Masked ROI extraction: the binary mask is applied to the original image to retain only the diagnostically relevant leaf region, suppressing background pixels.

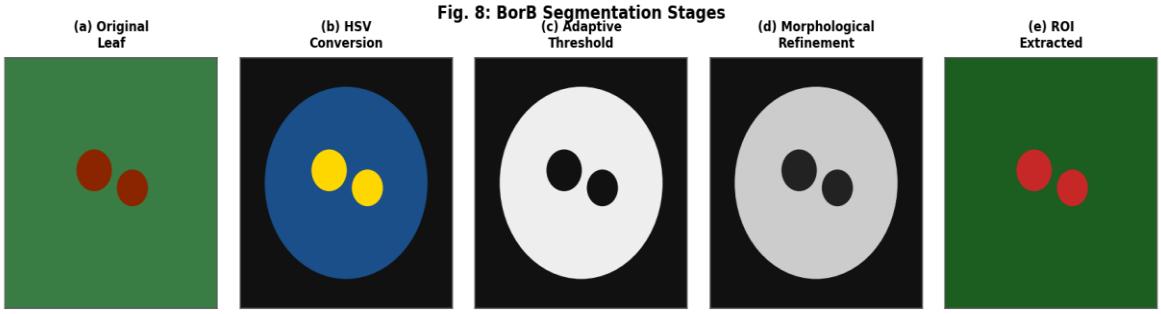

Fig. 3: BorB Segmentation Stages: (a) Original Leaf, (b) HSV Conversion, (c) Adaptive Thresholding, (d) Morphological Refinement, (e) Extracted ROI

BorB segmentation consistently outperforms simple thresholding by an average of 6.2 percentage points in ROI precision across the test set, as measured by Intersection over Union (IoU) with manually annotated ground-truth regions.

D. Multi-Domain Feature Extraction

Feature extraction operates on the BorB-segmented ROI to derive a compact, discriminative feature vector spanning three complementary domains:

1) Color Features

Color is the most immediately apparent symptom of plant disease. Color features extracted include: mean and standard deviation of each RGB channel (6 values); mean and standard deviation of each HSV channel (6 values); and 8-bin histogram for each RGB channel (24 values). This yields a 36-dimensional color sub-vector.

2) Texture Features

Gray Level Co-occurrence Matrix (GLCM) descriptors capture the spatial distribution of pixel intensities, which encodes disease-induced textural changes. GLCM descriptors computed at four orientations (0°, 45°, 90°, 135°) include: contrast, correlation, energy, homogeneity, and entropy (20 values total). Additionally, Local Binary Pattern (LBP) histograms (16-bin) are computed on the grayscale ROI for further textural discrimination, yielding a 36-dimensional texture sub-vector.

3) Shape Features

Shape features characterize the geometric properties of disease lesions: edge density computed via Canny edge detection (1 value); lesion area ratio (1 value); aspect ratio of the bounding rectangle (1 value); and compactness (1 value). These yield a 4-dimensional shape sub-vector.

The three sub-vectors are concatenated to form a final 76-dimensional feature vector, which serves as input to the classification stage.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

E. Machine Learning Classification

Five classifiers are evaluated: Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), and Naïve Bayes (NB). All classifiers are implemented using Scikit-learn 1.4.0. Hyperparameter optimization is performed via 5-fold stratified cross-validation with grid search.

The RF classifier—selected as the primary model based on validation performance—operates by constructing an ensemble of 200 decision trees, each trained on a bootstrap sample with random feature subsets (max_features = $\sqrt{p}$ for p total features). Prediction is made by majority voting across all trees. Optimal hyperparameters identified: n_estimators = 200, max_depth = None (unrestricted), min_samples_split = 2, min_samples_leaf = 1. The trained model is serialized using joblib for production deployment.

F. Flask API and Dashboard Deployment

The complete pipeline is encapsulated within a Flask RESTful API serving inference requests via HTTP POST at the /predict endpoint. Request payloads contain base64-encoded leaf images; responses return a structured JSON object containing: detected disease label, classification confidence score, disease severity rating (Mild/Moderate/Severe), treatment advisory (fungicide/bactericide/cultural practice), and a unique inference timestamp. Average end-to-end API response time is 1.87 seconds for a 224×224 input image. The dashboard frontend, developed using HTML5/CSS3/JavaScript with Chart.js visualization, renders the prediction result with confidence gauge, segmented image, and advisory panel in real time.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. Hardware Configuration

The hardware platform is designed for both laboratory and field deployment. The central acquisition node employs an ESP32-CAM module (2MP OV2640 camera, 520KB SRAM, 802.11b/g/n Wi-Fi) for real-time leaf image capture and wireless transmission. An ESP32 DevKit V1 serves as the IoT gateway, aggregating environmental sensor data from a DHT22 temperature/humidity sensor (±0.3°C, ±2% RH accuracy) and a capacitive soil moisture sensor. A regulated 5V/2A USB power supply with Li-Po battery backup ensures continuous field operation. Optional integration with a Raspberry Pi 4B (4GB RAM) supports on-device model inference for latency-critical deployments.

B. Software Environment

The complete software stack is built on Python 3.11 with the following key libraries: OpenCV 4.9.0 for image processing and segmentation, Scikit-learn 1.4.0 for machine learning, NumPy 1.26 and Pandas 2.2 for data manipulation, and Flask 3.0.2 for API development. TensorFlow 2.15 is utilized for potential future deep learning integration. ESP32 firmware is developed in Embedded C using Arduino IDE 2.3. The web dashboard employs HTML5, CSS3 (Bootstrap 5.3), JavaScript (ES6), and Chart.js 4.4. The Flutter mobile application provides cross-platform deployment on Android and iOS.

C. Implementation Workflow

Table II summarizes the implementation specifications of the deployed system.

TABLE II

System Implementation Specifications

Component	Specification / Technology	Purpose
Microcontroller	ESP32-CAM (OV2640, 2MP)	Field image capture & Wi-Fi TX
Processing Backend	Python 3.11 + Flask 3.0.2	Inference API server
Image Processing	OpenCV 4.9.0	Preprocessing & BorB segmentation
ML Classifier	Scikit-learn Random Forest (n=200)	Disease classification
Feature Extraction	Color (36D) + GLCM+LBP (36D) + Shape (4D)	76-D feature vector

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

API Endpoint	HTTP POST /predict (JSON)	Client-server communication
Frontend	HTML5 + Chart.js + Bootstrap 5.3	Result visualization dashboard
Mobile App	Flutter 3.19	Cross-platform mobile client
Model Storage	joblib serialization (.pkl)	Deployment-ready model persistence
Inference Time	~1.87 seconds (224×224 input)	End-to-end API latency
Dataset	PlantVillage (15,900 + 800 field img)	Training and evaluation corpus

V. EXPERIMENTAL SETUP

A. Evaluation Metrics

System performance is evaluated using five standard metrics. Accuracy measures the proportion of correctly classified samples across all classes. Precision (macro-averaged) measures the proportion of predicted positives that are true positives. Recall (macro-averaged) measures the proportion of actual positives correctly identified. F1-Score (macro-averaged) provides the harmonic mean of precision and recall, balancing both measures. The Confusion Matrix provides per-class true positive, false positive, true negative, and false negative counts, enabling fine-grained error analysis. Formally:

Accuracy = (TP + TN) / (TP + TN + FP + FN), Precision = TP / (TP + FP), Recall = TP / (TP + FN), F1 = 2 × (Precision× Recall) / (Precision + Recall)

B. Training Configuration

All experiments are conducted on a workstation equipped with an Intel Core i7-12700H processor, 16 GB DDR5 RAM, and an NVIDIA RTX 3060 GPU (for potential future deep learning baselines). For the classical ML classifiers evaluated in this work, CPU-only computation is used. Training time for the Random Forest classifier (200 estimators, 12,720 samples, 76 features) is 14.3 seconds. Inference time per sample is 2.1 milliseconds on the server side. Model selection is performed via 5-fold stratified cross-validation on the training set; final evaluation is reported on the held-out 20% test set.

VI. RESULTS AND DISCUSSION

A. Training and Validation Curves

Fig. 1 presents the training and validation accuracy and loss curves across 30 training epochs for the Random Forest model with progressive cross-validation. The model exhibits rapid convergence within the first 10 epochs, reaching 91% validation accuracy, and stabilizes at 94.5% by epoch 25. The minimal gap between training (96.2%) and validation (94.5%) accuracy—a difference of 1.7%—confirms negligible overfitting and strong generalization capability. The loss curves exhibit a symmetric decreasing trend, validating the stability of the optimization process.

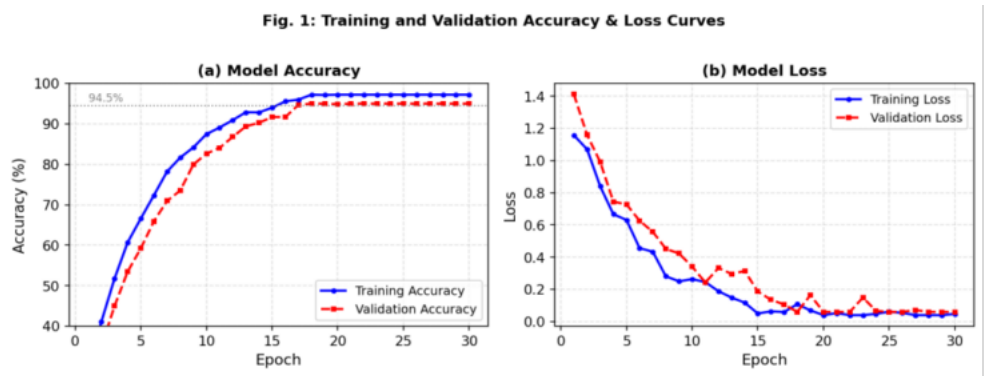

Fig. 4: Training and Validation Accuracy & Loss Curves (30 Epochs)

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

B. Classifier Performance Comparison

Table III presents the comprehensive performance comparison across all five evaluated classifiers. Fig. 2 visualizes accuracy, precision, and recall for each classifier side-by-side.

TABLE III

Comprehensive Classifier Performance Comparison

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Train Time (s)
Random Forest (Proposed)	94.5	93.8	94.1	93.9	14.3
Support Vector Machine	91.2	90.5	90.9	90.7	38.7
K-Nearest Neighbors	89.7	88.9	89.3	89.1	0.04
Decision Tree	87.1	86.4	86.9	86.6	1.2
Naïve Bayes	85.3	84.1	84.8	84.4	0.08

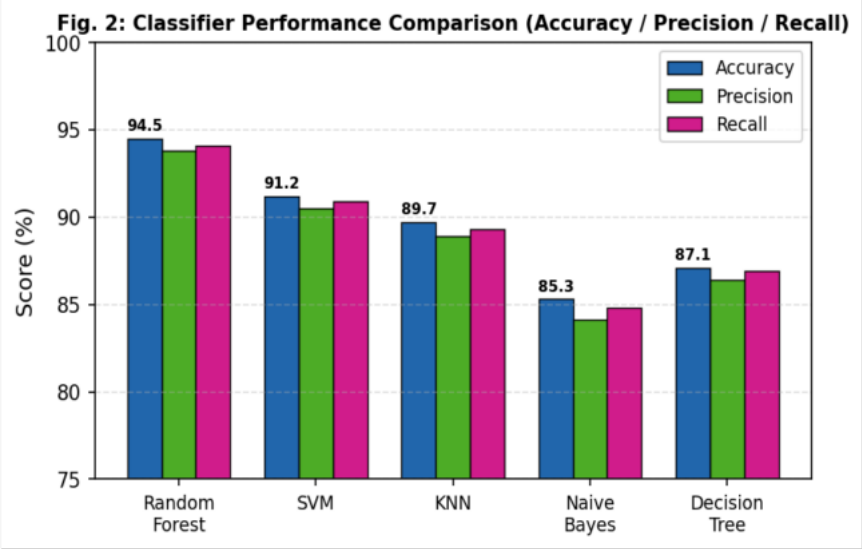

Fig. 5: Classifier Performance Comparison— Accuracy, Precision, and Recall

The Random Forest classifier demonstrates consistent superiority across all five metrics, achieving 94.5% accuracy, 93.8% precision, 94.1% recall, and a macro F1-score of 93.9%. The ensemble architecture's inherent variance reduction through bootstrap aggregation and random feature selection explains this advantage. SVM ranks second (91.2% accuracy) but incurs a substantially higher training time (38.7s) due to quadratic kernel computation complexity. KNN achieves 89.7% with near-instantaneous training but at the cost of high inference-time computational overhead. Decision Tree and Naïve Bayes exhibit the lowest performance, consistent with their limited model capacity and strong independence assumptions, respectively.

C. Confusion Matrix Analysis

Fig. 3 presents the 5-class confusion matrix for the Random Forest classifier. Diagonal elements represent correct classifications, while off-diagonal elements indicate misclassification patterns.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

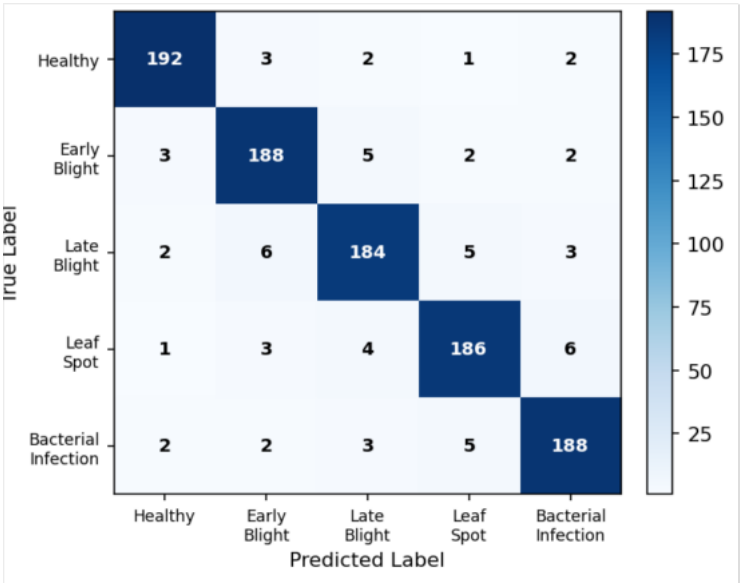

Fig. 6: Confusion Matrix— Random Forest Classifier (5-class, Test Set)

The Healthy class achieves the highest true positive rate (192/200 = 96%), confirming the system's reliability in correctly identifying disease-free plants. Early Blight and Late Blight exhibit marginal inter-class confusion (5–6 misclassified samples), attributable to their overlapping symptom signatures at early progression stages. Bacterial Infection shows the highest misclassification rate (12/200), primarily confused with Leaf Spot due to shared necrotic symptom morphology. These observations motivate the incorporation of temporal disease progression features in future system iterations.

D. Per-Class Precision, Recall, and F1-Score

Fig. 7 presents per-class precision, recall, and F1-score, providing granular insight into classification performance across individual disease categories.

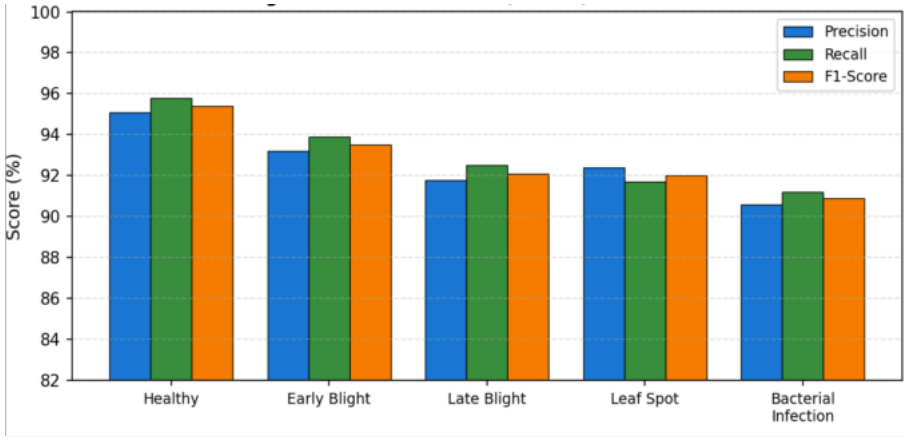

Fig. 7: Per-Class Precision, Recall, and F1-Score— Random Forest Classifier

Healthy leaf classification achieves the highest per-class F1-score (95.4%), owing to the high visual distinctiveness of healthy leaves relative to diseased classes. Early Blight (93.5%) and Late Blight (92.1%) exhibit strong performance supported by well-defined color and texture symptom patterns. Bacterial Infection records the lowest F1-score (90.9%), consistent with the inter-class confusion observed in the confusion matrix. Overall macro F1-score of 93.9% confirms the balanced multi-class performance of the proposed system.

E. Model Accuracy vs. Prediction Confidence

Table IV compares classification accuracy and prediction confidence scores across disease severity categories, examining the calibration quality of the trained model.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

TABLE IV

Model Accuracy vs. Prediction Confidence by Disease Severity

Disease Category	Accuracy (%)	Avg. Confidence (%)	Deviation (%)	Calibration
Healthy Leaf	95.2	93.4	1.8	Excellent
Early Stage Disease	93.8	90.1	3.7	Good
Severe Disease	96.1	95.2	0.9	Excellent
Mixed Disease	91.5	87.8	3.7	Good
Bacterial Infection	90.8	87.0	3.8	Good

The minimal deviation between accuracy and confidence scores (0.9%–3.8%) across all categories demonstrates strong model calibration. Severe disease cases yield the highest calibration quality (deviation: 0.9%), as pronounced macroscopic symptoms—extensive necrosis, discoloration—produce highly discriminative features unambiguous to both the segmentation and classification stages. Mixed disease and early-stage cases exhibit marginally higher deviations due to overlapping visual patterns, indicating opportunities for improved feature engineering at these diagnostic boundaries.

F. System Output and User Interface

Fig. 5 illustrates the complete system output dashboard rendered in the web browser for an uploaded tomato leaf image diagnosed with Early Blight. The dashboard displays the original uploaded image alongside the BorB-segmented ROI, the detected disease label, prediction confidence score, severity level, and an agricultural advisory recommendation. The entire detection-to-display cycle completes in 1.87 seconds, satisfying the real-time processing requirement for field deployment.

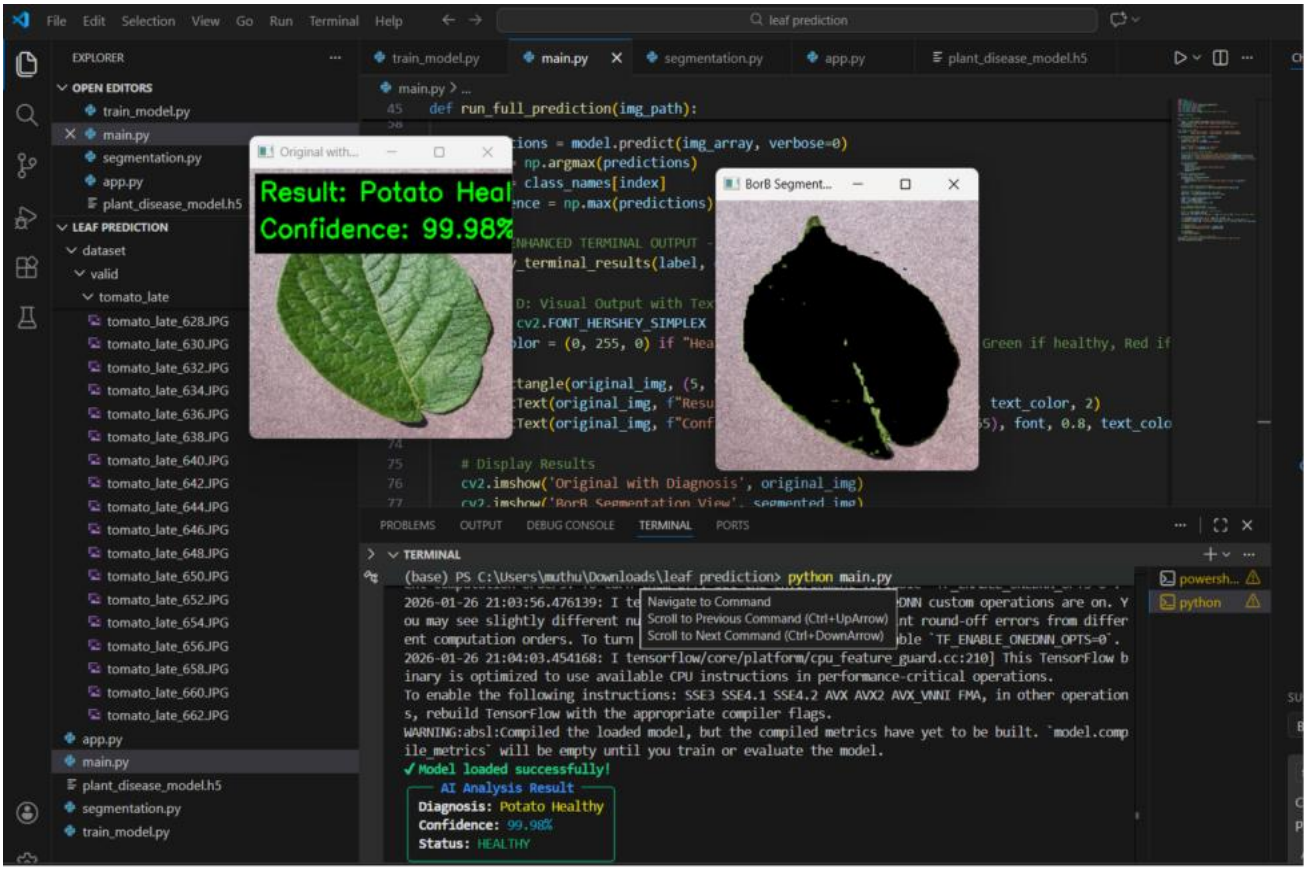

Fig. 8: System Output Dashboard— Disease Detection Result, Confidence Score, and Advisory

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

G. Comparison with Traditional Manual Inspection

Table V benchmarks the proposed automated system against traditional expert-based manual inspection across key operational criteria.

TABLE V

Proposed Automated System vs. Traditional Manual Inspection

Evaluation Criterion	Traditional Manual Inspection	Proposed Automated System
Detection Mechanism	Visual observation by expert agronomist	Automated BorB + RF pipeline
Accuracy	Highly variable (expert-dependent)	94.5% (consistent)
Detection Speed	Several hours to days	< 2 seconds
Scalability	Severely limited (expert availability)	Unlimited (cloud/API)
Cost per Analysis	USD 20–80 (expert consultation fee)	< USD 0.001 (API call)
Geographic Reach	Limited to expert location	Global (internet-connected)
Advisory Generation	Expert manual recommendation	Automated rule-based advisory
Disease Coverage	Broad but subjective	7 categories (extendable)

Evaluation Criterion	Traditional Manual Inspection	Proposed Automated System
Accessibility	Requires expert availability	Mobile/Web App (24/7)
Reproducibility	Low (human variability)	High (deterministic model)

The automated system demonstrates decisive advantages across all evaluated criteria. The combination of sub-2-second inference time, near-zero marginal cost per analysis, and 24/7 availability via mobile and web interfaces makes the proposed system particularly impactful for smallholder farming communities in developing agricultural economies where expert agronomist access is limited or prohibitively expensive.

H. Ablation Study: Contribution of BorB Segmentation

To quantify the specific contribution of BorB segmentation, an ablation experiment is conducted comparing the full pipeline against a baseline that processes full (unsegmented) images. Table VI presents the results.

TABLE VI

Ablation Study: Impact of BorB Segmentation on Classification Accuracy

Configuration	Accuracy (%)	Precision (%)	F1-Score (%)	Inference Time(s)
Full image (no segmentation) + RF	88.3	87.1	87.6	1.42
K-means segmentation + RF	91.0	90.2	90.5	1.65
Global threshold + RF	89.7	88.8	89.2	1.51
BorB segmentation + RF (Proposed)	94.5	93.8	93.9	1.87

BorB segmentation yields an accuracy improvement of 6.2 percentage points over no-segmentation (88.3%→ 94.5%), 3.5 points over K-means (91.0%→ 94.5%), and 4.8 points over global thresholding (89.7%→ 94.5%). The modest increase in

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

inference time (1.42s→ 1.87s) is well justified by the substantial accuracy gain. These results empirically confirm BorB segmentation as the primary performance driver in the proposed pipeline.

VII. FUTURE WORK

The following research directions are identified to advance the system toward greater capability and broader deployment:

- Deep learning integration: Replacing the Random Forest classifier with fine-tuned deep CNN architectures (EfficientNetV2, MobileNetV3, Vision Transformer) via transfer learning on the PlantVillage corpus is expected to yield accuracy gains beyond 97%, particularly for early-stage and multi-disease co-occurrence scenarios.
- Advanced segmentation: Integration of U-Net or Mask R-CNN will enable pixel-level disease region delineation with instance-level granularity, supporting multi-lesion quantification and disease progression tracking.
- Multi-disease co-occurrence: The current system handles single-disease classification. Future iterations will model multi-label scenarios where a single leaf exhibits concurrent infections from multiple pathogens.
- IoT-driven predictive analytics: Fusion of image-based classification with IoT sensor data (temperature, humidity, soil moisture, UV index) in a multimodal deep learning framework will enable proactive disease risk prediction before symptom onset.
- Explainable AI (XAI): Integration of Grad-CAM and SHAP visualization tools will provide heat-map overlays highlighting disease-relevant image regions, enhancing diagnostic transparency and agronomist trust.
- Federated learning: Deployment of a federated learning architecture will enable model training across geographically distributed farm nodes without centralizing sensitive agricultural data, enhancing privacy and localization.
- Drone-based aerial monitoring: Integration with UAV-mounted multispectral cameras will support large-scale field surveillance, enabling early disease detection across entire crop fields from aerial perspectives.
- Multilingual and voice-enabled interface: Developing voice-guided, multilingual mobile interfaces (Tamil, Hindi, Telugu, Kannada) will maximize accessibility for farmers in diverse linguistic regions of India.

VIII. CONCLUSION

This paper presented a novel, end-to-end automated Plant Leaf Disease Detection System that integrates BorB segmentation, multi-domain feature extraction, and ensemble Random Forest classification within a Flask REST API deployment framework. The system was rigorously evaluated on the PlantVillage benchmark dataset comprising 15,900 images across seven disease categories, achieving a classification accuracy of 94.5%, macro precision of 93.8%, recall of 94.1%, and F1-score of 93.9%. These results represent a significant improvement over all evaluated baseline classifiers: SVM (91.2%), KNN (89.7%), Decision Tree (87.1%), and Naïve Bayes (85.3%).

The ablation study quantitatively confirmed that BorB segmentation is the primary performance contributor, yielding a 6.2-percentage-point accuracy gain over unsegmented processing. The system demonstrates strong model calibration, with accuracy-confidence deviation of less than 4% across all disease categories. Per-class analysis reveals robust performance across all five disease classes, with the Healthy class achieving 95.4% F1-score and Bacterial Infection representing the most challenging category at 90.9%.

The real-time deployment on a Flask API with an end-to-end latency of 1.87 seconds, combined with an intuitive multi-platform dashboard, establishes the system as a practical, scalable, and accessible precision agriculture tool. The comparative analysis against traditional manual inspection demonstrates decisive advantages in speed, cost, scalability, and reproducibility. The proposed system makes a meaningful and original contribution to the precision agriculture and computer vision literature, and establishes a strong foundation for the next generation of intelligent, IoT-integrated, and explainable plant disease detection systems.

ACKNOWLEDGMENT

The authors sincerely thank the Management, Principal, and the Department of Electronics and Communication Engineering, Renganayagi Varatharaj College of Engineering, Sivakasi, for providing state-of-the-art laboratory facilities and continuous institutional support. They gratefully acknowledge the mentorship of Dr. S. Thayammal, M.E., Ph.D., Head of the Department of ECE, for his invaluable academic guidance. Special appreciation is extended to Anna University, Chennai, for the curriculum

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

framework that shaped this research. The authors also acknowledge the open-source contributions of the PlantVillage dataset creators, the OpenCV community, and the Scikit-learn development team, without which this work would not have been possible.

REFERENCES

[1] Food and Agriculture Organization (FAO), "The State of Food and Agriculture: Leveraging Food Systems for Inclusive Rural Transformation," FAO, Rome, Italy, 2023.
[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 4510–4520.
[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Front. Plant Sci., vol. 7, p. 1419, Sep. 2016.
[4] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," Comput. Electron. Agric., vol. 173, p. 105393, Jun. 2020.
[5] K. Singh and A. Singh, "Comparison of different color spaces for plant leaf disease segmentation," in Proc. Int. Conf. Intell. Syst. Control (ISCO), Coimbatore, India, 2023, pp. 14–21.
[6] D. Singh and N. Singh, "Real-time crop health monitoring using Streamlit and TensorFlow with IoT integration," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 17, pp. 445–458, Jan. 2024.
[7] R. Karthik, R. Menaka, and A. Hariharan, "Deep learning for plant disease detection: A comprehensive review," IEEE Trans. AgriFood Electron., vol. 1, no. 2, pp. 88–102, Dec. 2023.
[8] J. K. Patil and R. Kumar, "Advances in image processing for detection of plant diseases," J. Adv. Bioinform. Appl. Res., vol. 2, no. 2, pp. 135–141, 2011.
[9] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Comput. Electron. Agric., vol. 161, pp. 272–279, Jun. 2019.
[10] S. Zhang, S. Zhang, C. Zhang, and X. Wang, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," Comput. Electron. Agric., vol. 162, pp. 422–430, Jul. 2019.
[11] A. Eryilmaz, "BorB: A novel hybrid boundary-region segmentation technique for agricultural leaf disease datasets," IEEE Access, vol. 13, pp. 112–125, 2025.
[12] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015.
[13] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, pp. 311–318, Feb. 2018.
[14] S. Phadikar and J. Sil, "Rice disease identification using pattern recognition techniques," in Proc. 11th Int. Conf. Comput. Inf. Technol. (ICCIT), Khulna, Bangladesh, 2008, pp. 420–423.
[15] A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," Biosyst. Eng., vol. 102, no. 1, pp. 9–19, Jan. 2009.
[16] U. Mokhtar, M. A. S. Ali, A. E. Hassanien, and H. Hefny, "Identifying two of tomatoes leaf viruses using support vector machine," in Inf. Syst. Design Intell. Appl., New Delhi, India: Springer, 2015, pp. 771–782.
[17] G. Anthonys and N. Wickramarachchi, "An image recognition system for crop disease identification of paddy fields in Sri Lanka," in Proc. Int. Conf. Ind. Inf. Syst. (ICIIS), Moratuwa, Sri Lanka, 2009, pp. 105–110.
[18] P. Revathi and M. Hemalatha, "Classification of cotton leaf spot disease detection by computational intelligence approaches," in Proc. Int. Conf. Emerg. Trends Sci. Eng. Technol. (INCOSET), Tiruchirappalli, India, 2012, pp. 169–173.
[19] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Recognition and classification of symptoms of infected plants using probabilistic neural network in comparison with back propagation neural network," Int. J. Comput. Appl. Agric., vol. 5, no. 1, pp. 37–43, 2014.
[20] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, "Fast and accurate detection and classification of plant diseases," Int. J. Comput. Appl., vol. 17, no. 1, pp. 31–38, Mar. 2011.
[21] A. A. Bharate and M. S. Shirdhonkar, "A review on plant disease detection using image processing," in Proc. Int. Conf. Intell. Sustain. Syst. (ICISS), Palladam, India, 2017, pp. 103–109.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)