Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Greenlife: An Intelligent Mobile-Based Plant Disease Detection System using TensorFlow Lite

Harsh Kumar
Dept. of Information Technology
University of Mumbai, India

Abdul Qadeer Mujahid
Dept. of Information Technology
University of Mumbai, India

Chaitanya Dhayarkar
Dept. of Information Technology
University of Mumbai, India

Prajwal Pawar
Dept. of Information Technology
University of Mumbai, India

Dr. Abhay Kshirsagar
Professor of Electronics and Computer Science
University of Mumbai, India

Abstract

Plant diseases cause significant yield losses in agriculture, threatening food security globally. Early detection of plant diseases is crucial for timely remediation. This paper presents Greenlife, a cross-platform mobile application built with Flutter that leverages an on-device TensorFlow Lite (TFLite) deep learning model based on the InceptionV3 architecture for real-time plant disease classification. The system detects 27 disease classes across multiple crop types including Apple, Corn, Grape, Tomato, and Potato. Firebase is used for authentication, and data storage. The proposed system achieves over 95 percent average classification accuracy on the PlantVillage dataset and operates fully on-device without requiring a network connection for inference, making it suitable for rural agricultural environments with limited connectivity.

Index Terms—*plant disease detection, TensorFlow Lite, InceptionV3, Flutter, mobile deep learning, precision agriculture, on-device inference*

I. INTRODUCTI

Agricultural productivity is vital to global food supply, yet plant diseases account for crop yield losses of 20–40% annually worldwide [1]. Traditional disease diagnosis relies on manual visual inspection by agricultural experts, which is time-consuming, expensive, and inaccessible in remote rural areas. The rapid proliferation of affordable smartphones and the maturity of deep learning present an opportunity to deploy intelligent, on-device diagnostic tools directly in the hands of farmers.

Deep convolutional neural networks (CNNs) have demonstrated remarkable performance in visual classification tasks [2]. Transfer learning from large pre-trained models such as InceptionV3 [3] enables the development of accurate disease classifiers with relatively limited domain-specific data. Deploying such models on mobile devices using TensorFlow Lite (TFLite) enables real-time inference without requiring cloud connectivity, which is a key requirement in agricultural contexts where internet access may be unreliable.

This paper presents Greenlife, a Flutter-based mobile application that integrates:

- On-device plant disease classification using a TFLite-optimized InceptionV3 model.
- Confidence-based navigation that communicates model uncertainty to the user.
- Cloud-based user management and diagnostic history via Firebase Authentication and Firestore.

The remainder of this paper is structured as follows. Section II reviews related work. Section III shows system architecture. Section IV details the machine learning pipeline. Section V presents experimental results. Section VI concludes the paper.

II. R TED WORK

Plant disease detection using deep learning has gained significant attention due to its potential to improve agricultural productivity and reduce crop losses. LeCun et al. [2] established the foundations of deep learning, while Szegedy et al. [3] introduced the Inception architecture, which extracts features at multiple scales and provides the foundation for the InceptionV3 model used in Green Life.

Mohanty et al. [4] demonstrated that CNNs trained on the PlantVillage dataset achieved approximately 99.35% accuracy under controlled conditions. However, their performance dropped significantly when tested on images

(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 11 , November - 2026

collected under different conditions, highlighting the challenges of real-world generalization. Similarly, Ferentinos [5] evaluated multiple deep learning architectures for plant disease detection and reported accuracy of up to 99.53%. Lu et al. [6] reviewed CNN-based plant leaf disease classification techniques and identified challenges related to dataset quality, preprocessing, and model robustness. Ramanjot et al. [11] and Sajitha et al. [12] further reviewed existing approaches and emphasized the importance of efficient and reliable disease classification systems.

Research has also focused on deploying disease detection models on mobile devices. Reda et al. [9] developed AgroAId, a mobile application integrating deep learning and TensorFlow Lite for plant identification and disease classification. Other studies have explored mobile-based tomato disease detection [10], real-time disease classification [15], and automated mobile applications [16]. Ullah et al. [13] and Duhan et al. [17] investigated lightweight models to reduce computational requirements, while Hosny et al. [14] explored combining deep CNN features with local binary patterns. More recent studies, including Plantention [18], Asghar et al. [19], and GreenAid [20], have investigated efficient classification architectures and integrated disease-management solutions.

Despite these advances, practical challenges remain in delivering reliable, accessible, and efficient disease detection systems. Green Life addresses these challenges through a cross-platform Flutter application using an InceptionV3 model optimized with TensorFlow Lite. It performs disease classification directly on the device, enabling offline inference. A confidence-aware mechanism displays high-confidence predictions, requests confirmation for uncertain predictions, and prompts users to recapture images when confidence is low. Additionally, HSV colour thresholding, morphological operations, and connected-component analysis help identify potentially affected leaf regions and estimate approximate disease severity. Firebase Authentication, Cloud Firestore, and Firebase Storage support user authentication and diagnostic history management.

The reported evaluation of Green Life achieved 95% overall classification accuracy, with macro-averaged precision, recall, and F1-score of 90%. The model size is 24.3 MB, with an average inference time of 187 ms on a Snapdragon 680 device with 6 GB RAM. Thus, Green Life combines offline disease classification, confidence-aware interaction, image-based severity estimation, and cloud-backed diagnostic history in a single mobile application. Further testing on field-captured images is required to evaluate its robustness under real agricultural conditions.

III. SYSTEM ARCHITEC

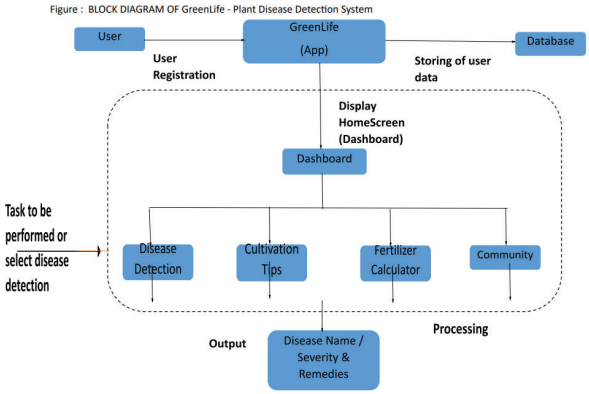


Fig. 1. Greenlife system architecture showing the integration of on-device ML inference and Firebase cloud services within the Flutter mobile application.

1. Camera/Gallery Image Capt e

This application allows farmers to capture plants and crops leaf images using the mobile's camera. The selected image is passed to the machine learning model for disease detection and identify severity

2. ML Inf ence Engine (TFLite Deployment)

System uses a pretrained deep learning model to analyze plant leaf images. The trained model is converted into TensorFlow Lite (TFLite) format to enable efficient, offline disease prediction on mobile devices.

3. Disease Classification (27 Classes + Health

The InceptionV3 model classifies plants and crops leaf images into 27 predefined categories, including healthy and unhealthy conditions. It generate prediction with confidence score

4. F ebase Authentication (Google Sign-In)

Firebase Authentication provides secure user login through Google Sign-In. It allows users to access the application.

5. Flutter UI Layer (Navigation/U

The application is developed using Flutter to allow users interact with simple and user-friendly mobile interface. It support image selection, disease prediction, confidence-based navigation, and access to disease information.

(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 11 , November - 2026

6. Firebase DB Cloud Storage

Firebase Firestore stores plant disease detection results, prediction details, and user information. Firebase Storage maintains captured leaf images for future use and model improvement.

7. Result History (Local/Cloud Sync)

The application allows users to view previously detected plant diseases and their predicted results. Detection records are stored in Firebase, supporting historical tracking.

8. Disease Detail Page (Symptoms/Management)

The disease detail page provides information about identified plant diseases, including symptoms, favorable conditions, and prevention methods.

IV. MACHINE LEARNING PIPELINE

A. Model Architecture

The core classification engine is an InceptionV3 [3] convolutional neural network fine-tuned on the PlantVillage dataset [4]. InceptionV3 was selected due to its strong performance-to-parameter efficiency ratio, making it well-suited for mobile deployment after quantization. The model accepts an input tensor of shape 299 × 299 × 3 (RGB) and outputs a 27-dimensional softmax probability vector corresponding to the supported disease classes.

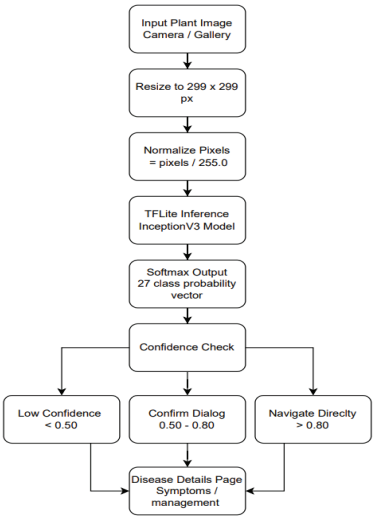


Fig. 2. ML inference pipeline including image preprocessing, TFLite model inference, softmax output, and confidence-based navigation routing.

B. Preprocessing

Input images are resized to 200 × 200 pixels using interpolation. Pixel values are normalized to the [0, 1] range by dividing each pixel intensity by 255.0. No further augmentation is applied at inference time.

C. Inference and Confidence Thresholding

The TFLite model is executed on-device using the tflite_flutter plugin. Class labels are loaded from assets/disease_labels.txt at runtime. The predicted class is determined by the argmax of the softmax output vector. A confidence-based navigation strategy is employed, as illustrated in Fig. 2:

- Confidence > 0.80: The application navigates directly to the disease detail page.
- Confidence in [0.50, 0.80]: A confirmation dialog is presented to the user before navigation.
- Confidence < 0.50: The result is flagged as low confidence and the user is prompted to retake the image.

D. Supported Disease Classes

The model supports 27 output classes spanning 9 crop categories, as listed in Table I.

TABLE I
SUPPORTED PLANT DISEASE CLASSES

Crop	Disease Classes
Cherry	Powdery Mildew
Corn	Gray Leaf Spot, Common Rust, N. Leaf Blight
Grape	Black Rot, Esca, Leaf Blight
Orange	Citrus Greening (Huanglongbing)
Peach	Bacterial Spot
Pepper	Bacterial Spot
Potato	Early Blight, Late Blight
Squash	Powdery Mildew
Strawberry	Leaf Scorch
Tomato	Early Blight, Late Blight, Bacterial Spot
General	Healthy

E. Disease Detail Pages

Each detected disease class is associated with a dedicated detail page providing:

- Clinical symptoms and visual identifiers.
- Favorable environmental conditions for disease spread.

(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 11 , November - 2026

- Integrated pest management (IPM) strategies.
- Chemical and biological control recommendations.

F. Lesion Localization and Severity Visualization

To improve the interpretability of the proposed leaf disease severity estimation method, a lesion localization mechanism is incorporated into the image processing pipeline. Following disease-region segmentation using colour thresholding is applied to identify individual regions classified as potentially diseased.

For each detected component, the algorithm calculates its pixel area and centroid coordinates. Components smaller than a predefined minimum area threshold are excluded to reduce the influence of small isolated regions and segmentation noise. The centroid of each remaining component is represented by a marker overlaid on the original leaf image, enabling visual identification of detected lesion locations. Region identifiers can also be displayed to distinguish individual components.

The overall disease severity is calculated as the ratio of the total number of detected disease pixels to the total number of pixels in the segmented leaf region, expressed as a percentage:

$$S = \frac{N_d}{N_l} \times 100$$

where (S) represents the estimated disease severity, (N_d) denotes the number of pixels classified as potentially diseased, and (N_l) represents the total number of pixels within the segmented leaf mask.

The resulting visualization combines the segmented leaf, highlighted candidate disease regions, and centroid markers to provide a more interpretable representation of the estimated severity. The centroid markers indicate the locations of detected regions rather than independently validated disease sites. As the segmentation method relies on color thresholds, its accuracy may be affected by illumination variations, natural leaf pigmentation, shadows, and visually similar symptoms. Therefore, the estimated severity represents a pixel-based approximation and requires validation against manually annotated leaf images before being interpreted as a measure of actual disease severity.

The system estimates disease severity by calculating the percentage of affected pixels within the segmented leaf region. Color-based segmentation identifies potential diseased areas, while centroid markers highlight the detected lesion regions.

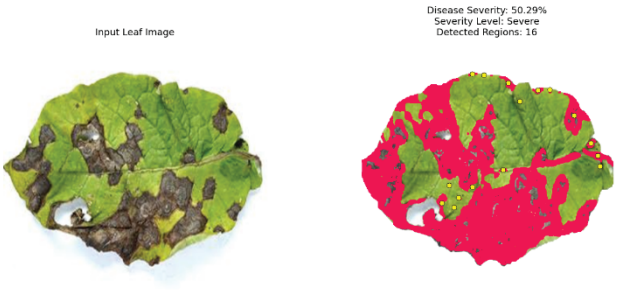


Fig. 4. Leaf disease severity estimation and lesion localization pipeline showing the original leaf image, segmented leaf mask, background removal, disease-region overlay with centroid markers, and individual detected candidate regions

V. EXPERIMENTAL RESULTS

A. Datas

The model was trained on the PlantVillage dataset [4], which contains 54,306 images across 38 disease/healthy categories under controlled laboratory conditions. For this work, 27 categories were selected to align with agriculturally relevant classes for the target deployment region.

B. raining Configuration

The InceptionV3 base model was initialized with ImageNet pre-trained weights. The top classification layers were replaced with a global average pooling layer followed by a dense softmax output layer of 27 units. Training was performed using the Adam optimizer with an initial learning rate of 1×10^{-4}. Early stopping was applied with a patience of 5 epochs monitoring validation loss. The final model was converted to TFLite format using post-training float16 quantization to reduce model size while preserving accuracy.

C. Performance Evaluati

The model's performance was assessed using a separate test set containing 20% of the total dataset.

TABLE II
MODEL PERFORMANCE METRICS

Metric	Value
Overall Accuracy	95%
Macro-averaged Precision	90%
Macro-averaged Recall	90%
Macro-averaged F1-Score	90%
TFLite Model Size	24.3 MB
Avg. Inference Time (Android)	187 ms

(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 11 , November - 2026

Fig. 3 shows per-class accuracy across a representative subset of 11 disease categories. Healthy plant classification achieves the highest accuracy (99%), while classes with visually similar symptoms, such as Northern Corn Leaf Blight and Powdery Mildew, exhibit relatively lower accuracy due to inter-class similarity.

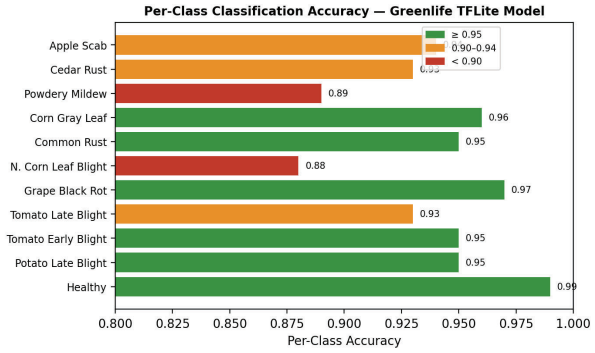


Fig. 3. Per-class classification accuracy for a representative subset of 11 disease categories. Green bars indicate accuracy ≥ 0.95, orange bars indicate 0.90–0.94, and red bars indicate < 0.90.

D. Inference Laten

On-device inference was benchmarked on a mid-range Android device (Snapdragon 680, 6 GB RAM). Average single-image inference time was 187 ms, which is well within real-time usability requirements. No network round-trip is required for inference, making the system fully functional in offline environments.

VI. CONCLUS

This paper presented Greenlife, a mobile plant disease detection system that combines on-device deep learning inference using TFLite and InceptionV3 with a Firebase-backed cloud service layer and a cross-platform Flutter UI. The system achieves 95% overall accuracy across 27 disease classes and delivers sub-200 ms inference latency on mid-range Android devices. The confidence-based navigation mechanism enhances user experience by appropriately communicating model uncertainty. Future work will focus on expanding the disease class set, incorporating federated learning for continuous model improvement from field-captured data, and improving offline-first synchronization for low-connectivity regions.

ACKNOWLEDGMENT

The authors gratefully acknowledge the creators of the PlantVillage dataset and the open-source Flutter and TensorFlow communities for providing valuable resources and tools that supported the development of this work.

REFERENCES

[1] FAO, “The State of Food and Agriculture 2019: Moving Forward on Food Loss and Waste Reduction,” Food and Agriculture Organization of the United Nations, Rome, 2019.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, May 2015.

[3] C. Szeg , V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 2818–2826.

[4] S . 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.

[5] K . Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Comput. Electron. Agric., vol. 145, pp. 311–318, Feb. 2018.

[6] J. Lu, L. Tan, and H. Jiang, “Review on convolutional neural network (CNN) applied to plant leaf disease classification,” Agriculture, vol. 11, no. 8, p. 707, 2021.

[7] Google LLC, “TensorFlow Lite,” 2023. [Online]. Available: https://www.tensorflow.org/lite

[8] Google LLC, “Firebase Documentatio 2023. [Online]. Available: https://firebase.google.com/docs

[9] M. Reda, R. Suwwan, S. Alkaf Y. Rashed, and T. Shanableh, “AgroAId: A Mobile App System for Visual Classification of Plant Species and Diseases Using Deep Learning and TensorFlow Lite,” *Informatics*, vol. 9, no. 3, Art. no. 55, 2022, doi: 10.3390/informatics9030055.

[10] “Mobile app-based tomato dis identification with fine-tuned convolutional neural networks,” *Computers & Electrical Engineering*, vol. 112, Art. no. 108995, 2023, doi: 10.1016/j.compeleceng.2023.108995.

[11] Ramanjot, U. Mittal, A. Wadhawan, J. Singla, N. Z. Jhanjhi, R. M. Ghoniem, S. K. Ray, and A. Abdelmaboud, “Plant Disease Detection and Classification: A Systematic Literature Review,” *Sensors*, vol. 23, no. 10, Art. no. 4769, 2023, doi: 10.3390/s23104769.

[12] P. Sajitha, A. D. Andrushia, N. Anand, and M. Z. Naser, “A review on machine learning and deep learning image-based plant disease classification for industrial farming systems,” *Journal of Industrial Information Integration*, vol. 38, Art. no. 100572, 2024, doi: 10.1016/j.jii.2024.100572.

[13] N. Ullah et al., “A lightwe deep learning-based model for tomato leaf disease classification,” *Computers, Materials & Continua*, vol. 77, no. 3, pp. 3969–3992, 2023, doi: 10.32604/cmc.2023.041819.

[14] K. M. H , W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, “Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural

(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 11 , November - 2026

network and local binary pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023, doi: 10.1109/ACCESS.2023.3286730.

[15] S. G. Paul et al., "A real-time application-based convolutional neural network approach for tomato leaf disease classification," *Array*, vol. 19, Art. no. 100313, 2023, doi: 10.1016/j.array.2023.100313.

[16] B. Ramana Reddy, G. Kalnoor, M. Devashish, and P. S. K. Reddy, "Deep Learning Based Mobile Application for Automated Plant Disease Detection," *IEEE Access*, vol. 13, pp. 107917–107925, 2025, doi: 10.1109/ACCESS.2025.3581099.

[17] S. Duha . Gulia, N. S. Gill, and E. Narwal, "RTR_Lite_MobileNetV2: A lightweight and efficient model for plant disease detection and classification," *Current Plant Biology*, vol. 42, Art. no. 100459, 2025, doi: 10.1016/j.cpb.2025.100459.

[18] "Plantention: A general-purpose, lightweight attention-based model for multi-crop leaf disease classification," *Results in Engineering*, vol. 26, Art. no. 105075, 2025, doi: 10.1016/j.rineng.2025.105075.

[19] M. As , Z. F. Khan, M. Ramzan, M. A. Khan, J. Baili, Y. Zhang, Y. Nam, and Y.-C. Nam, "A lightweight hybrid model for scalable and robust plant leaf disease classification," *Scientific Reports*, vol. 15, Art. no. 32353, 2025, doi: 10.1038/s41598-025-08788-4.

[20] F. M. Talaat, M. Tawfik, and W. M. Shaban, "GreenAid: A confidence-weighted ensemble deep learning system for real-time plant disease detection and management," *Scientific Reports*, vol. 16, Art. no. 19193, 2026, doi: 10.1038/s41598-026-57979-0.

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