OPEN ACCESS JOURNALS
Indiana Journal of Multidisciplinary Research
Abbreviated Tittle- Ind J Multi Res
ISSN *(Online)*- 2583-3820
Journal Homepage Link- https://indianapublications.com/Journals/IJMR
DOI: https://doi.org/10.5281/zenodo.21600299

INDIANA PUBLICATIONS
PRODUCTIVE AND QUALITY RESEARCH

Research Article

Volume-06|Issue-04|2026

Leaf Lens: Tomato Leaf Disease Detection

Dr. Sunita Chalageri[1], Sai Deeksha D[2], Sanjana S Tigadi[3], T Veneela[4], Varsha S N[5]

[1,2,3,4,5] K.S. Institute of Technology, Affiliated to VTU, Department of Computer Science and Engineering, Bangalore, India.

Article History
Received: 05.05.2026
Accepted: 22.06.2026
Published: 25.07.2026

Citation
Chalageri, S., Deeksha, S. D., Tigadi, S. S., Veneela, T. & Varsha, S. N. (2026). Leaf Lens: Tomato Leaf Disease Detection. *Indiana Journal of Multidisciplinary Research, 6*(4), 157-163.

Abstract: The productivity of tomato crops is negatively impacted by a number of leaf diseases, including Yellow Leaf Curl Virus, Septoria Leaf Spot, Early Blight, and Late Blight. In rural areas, manual disease detection is frequently unreliable and unavailable. This study presents LeafLens, a lightweight hybrid tomato leaf disease detection framework integrating CNN-based learning with statistical texture descriptors for practical agricultural deployment with 93.5%. The model uses deep learn-ing and lightweight classifiers to classify leaf images after extracting spatial, colour, and texture features. The scalable, affordable, and user-friendly design of LeafLens helps to increase agricultural sustainability and productivity.

Keywords: Artificial Intelligence (AI), Deep Learning, Diagnostic Accuracy, Deep Learning, Oral Squamous Cell Carcinoma (OSCC)

INTRODUCTION

Tomatoes are important for global agriculture, but are vulnerable to many leaf diseases. Traditional methods for detecting these diseases are based on visual inspections. These inspections can be inaccurate, take a long time and are often not possible in remote areas. Early and precise diagnosis is crucial for reducing yield loss. LeafLens uses digital image processing and CNNs to automate disease detection. Convolutional, ReLU activation, flattening, and fully connected layers are among the layers the model uses to process the input image, as shown in Fig. 1. The system tries to present an efficient and affordable method for farmers to diagnose diseases with minimal technical knowledge. LeafLens offers non-invasive, scalable, and accurate tomato leaf disease detection, which helps to enhance agricultural productivity and sustainability.

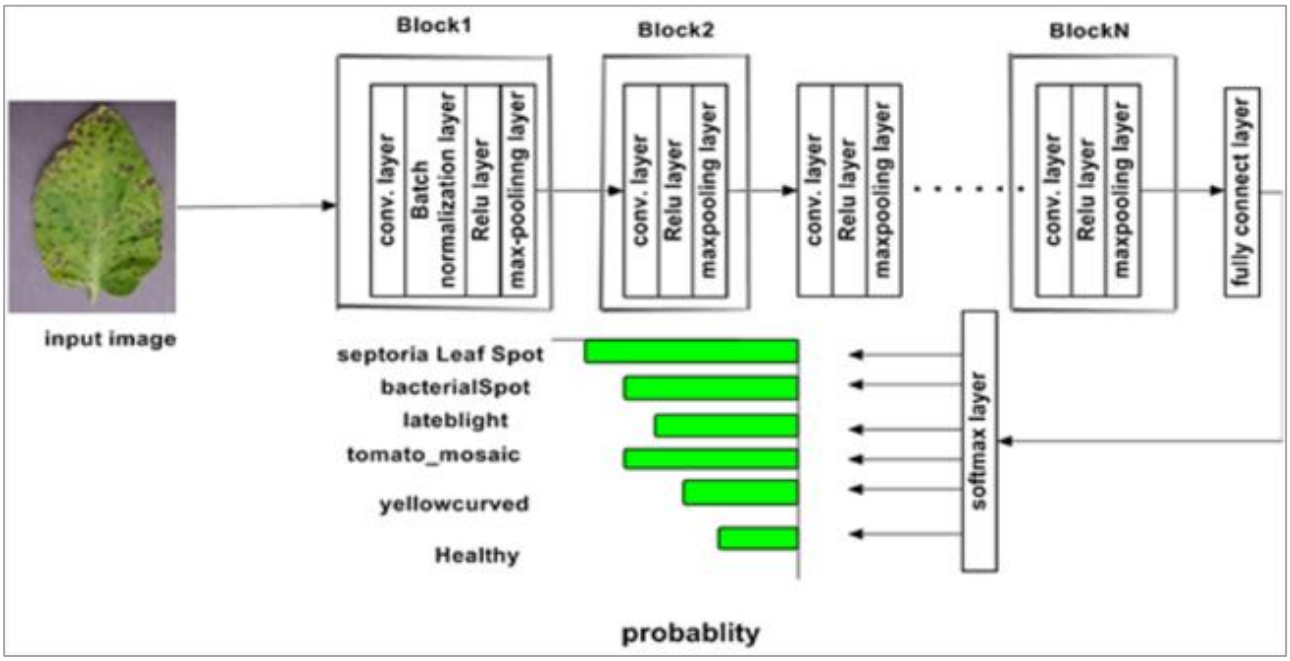

Fig. 1: CNN Architecture Flowchart

***Corresponding Author:** Sai Deeksha D

LITERATURE REVIEW

Many studies have looked at tomato disease detection through image processing:

1. Various machine learning and deep learning methods have been tried for the detection of plant diseases, particularly in tomato leaves. In a comparative study by Vishnu S. Babu et al. [1], various traditional models such as SVM, KNN, and Decision Tree were tried on plant leaf data. This motivated us to experiment and compare traditional methods in the initial stage of our project before settling down to a deep learning-based solution.
2. Balafas et al. [2] presented an elaborate overview of machine learning and deep learning methods in precision agriculture, including plant disease identification. Their algorithm classification informed our approach, particularly in choosing a hybrid feature-based and deep learning pipeline. Their focus on combining handcrafted features with CNNs validated our use of GLCM and LBP in addition to learned features.
3. Ahmed and Biradar [3] introduced a deep residual dense network specifically designed for the classification of tomato leaf diseases with high accuracy through residual connections and dense blocks. We similarly utilized residual learning methods to boost feature propagation and alleviate vanishing gradients in our CNN model.
4. Rashid Khan et al. [4] integrated K-means segmentation with GLCM and SIFT features followed by SVM classification. This work affected the image preprocessing and feature extraction pipeline used in our project. We utilized K-means for segmentation of regions and applied GLCM for texture feature extraction to aid both classical and deep learning models.
5. Hosny et al. [5] suggested the merging of deep CNN features with traditional texture descriptors such as LBP. We adopted a similar hybrid method in our work by merging CNN features with statistical descriptors (GLCM and LBP) to improve classification results. Our findings justified our choice of merging handcrafted and deep features rather than using CNN exclusively.

RESEARCH GAP AND CONTRIBUTIONS

Extensive research has been conducted in the area of plant and tomato leaf disease detection using both deep learning models and traditional handcrafted feature-based approaches. Convolutional Neural Networks (CNNs) have been widely adopted for automatic feature extraction and classification, particularly using datasets such as PlantVillage. Similarly, texture descriptors such as GLCM, LBP, HOG, and segmentation techniques like K-means clustering have been frequently employed in classical machine learning pipelines.

Although these methods have individually demonstrated promising results, many existing studies primarily focus on improving classification accuracy under controlled experimental conditions. Comparatively fewer works examine the systematic integration of lightweight CNN models with statistical texture features while also evaluating computational feasibility for deployment on resource-constrained devices.

In this context, the objective of the present work is not to introduce a fundamentally new CNN architecture or a novel feature extraction formulation. Instead, this study aims to:

- Develop a lightweight hybrid CNN–texture feature framework for tomato leaf disease detection.
- Examine the effectiveness of combining learned deep features with statistical texture descriptors.
- Evaluate classification performance in relation to computational efficiency.

DATASET DESCRIPTION

Images of plant leaves from six different classes: Healthy, Early Blight, Late Blight, Mosaic Virus(Fig. 3), Septoria Leaf Spot, Leaf Mold and Yellow Leaf Curl Virus (Fig. 2) make up the dataset.

To ensure uniformity and lower computational complexity, every image is resized to 256x256 pixels. There are three sections to the dataset: The model is trained using the Training Set (0.32 GB). It includes a wide variety of labeled photos showing different disease stages and leaf conditions. Testing Set (0.32 GB): Used to test the model's ability to general-ize by analyzing its performance on unknown data. A smaller subset known as the validation set is used during training to adjust hyperparameters and avoid overfitting. A strong basis for developing a trustworthy deep learning model for the classification of plant diseases is provided by this structured dataset.

Fig. 2: Yellow Curl Disease

***Corresponding Author:** Sai Deeksha D

Fig. 3: Mosaic Virus

Fig. 4: Grayscale Conversion

PROPOSED METHODOLOGY

The methodology does not propose a new CNN architecture; instead, it integrates established feature extraction techniques with a lightweight CNN framework to evaluate performance efficiency trade-offs in real-world agricultural scenarios. The steps involved are as follows:

Sl. No.	Class Name	Number of Images
1	Healthy	1,591
2	Early Blight	1,000
3	Late Blight	1,909
4	Leaf Mold	952
5	Septoria Leaf Spot	1,771
6	Yellow Leaf Curl Virus	5,357
7	Mosaic Virus	373
Total		**12,953**

1. Image Acquisition: Images are captured using cameras or smartphones under various lighting conditions.
2. Preprocessing: This step enhances image quality and reduces noise to improve feature extraction and classification performance.

– Grayscale Conversion: Converts the RGB image to grayscale to reduce complexity. The below figure shows the Grayscale conversion in Fig. 4

3. Feature Extraction: In this phase, significant features are extracted from the segmented regions for classification as shown in Fig. 6 and Fig. 7. Color Features (Mean and Standard Deviation) Statistical measures of color intensi-ties in the region:

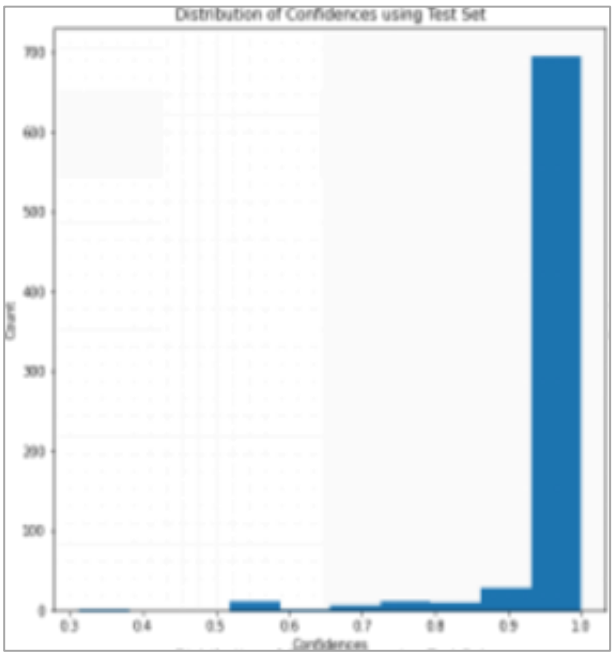

Fig. 6: Histogram

***Corresponding Author:** Sai Deeksha D

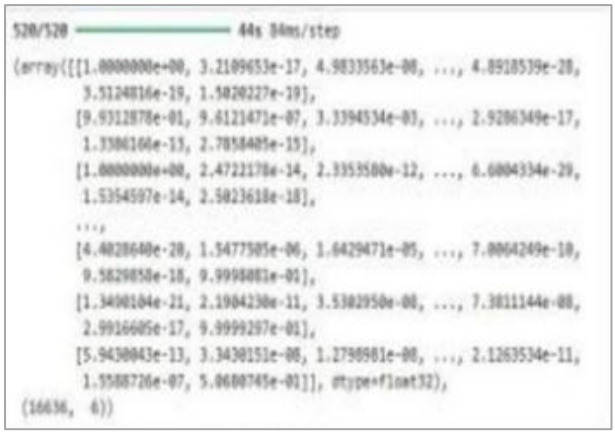

Fig. 7: Feature vectors

– GLCM Features (Gray-Level Co-occurrence Matrix) These capture tex-ture information by measuring how often pairs of pixel intensities occur.
– Binary Pattern (LBP) captures local texture patterns by comparing each pixel with its neighbors, while Histogram of Oriented Gradients (HOG) describes the edge orientation and shape information. Both are useful for characterizing leaf textures and disease patterns.

4. Segmentation: Segmentation isolates relevant regions, such as diseased parts of the leaf, for focused analysis.
– Otsu's Thresholding: Automatically determines the optimal threshold to separate foreground and background represented in Fig. 8.

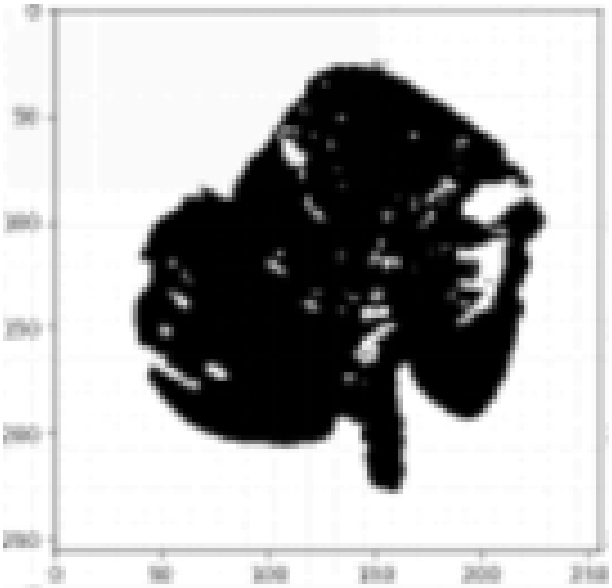

Fig. 8: Otsu thresholding

– K-Means Clustering: Divides image pixels into *k* clusters based on simi-larity (e.g., color or intensity) as shown in Fig. 9

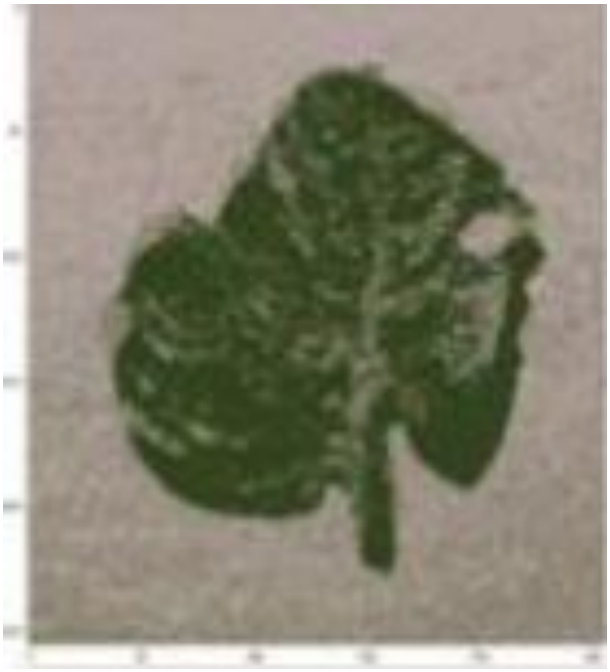

Fig. 9: K-means clustering

– Canny Edge Detection: Detects edges using gradient magnitude and di-rection. The Canny Edge Output is shown in Fig. 10

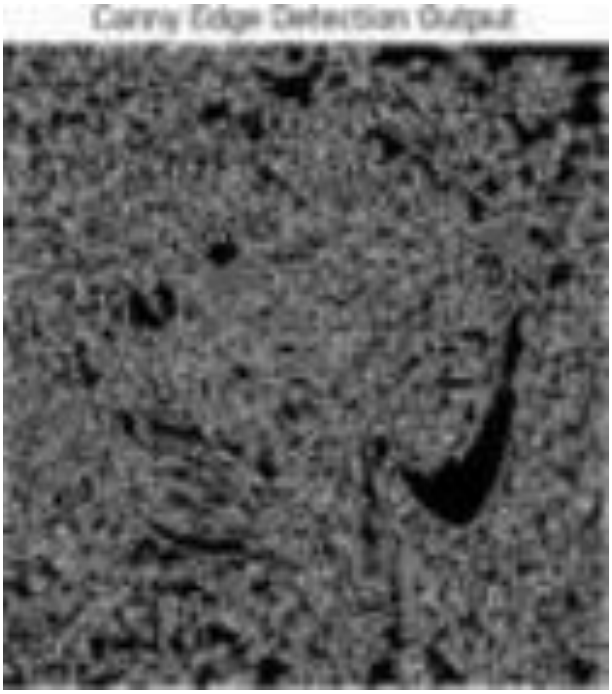

Fig. 10: Canny Edge output

5. Classification: Based on extracted features, classify using rule-based logic or light ML (e.g., SVM, Decision Trees) the classification report is shown in Fig. 11.

***Corresponding Author:** Sai Deeksha D

	Precision	Recall	F1
Bacterial-spot	100.00%	99.08%	99.54%
Early-blight	95.65%	98.88%	97.24%
Healthy	98.29%	100.00%	99.14%
Late-blight	95.33%	98.08%	96.68%
Leaf-mold	98.26%	95.76%	97.00%
Mosaic-virus	100.00%	100.00%	100.00%
Septoria-leaf-spot	97.59%	93.10%	95.29%
Yellow-leaf-curl-virus	100.00%	100.00%	100.00%

Fig. 11: Classification Report

6. Final Classification Framework Clarification:
– Even though the traditional classifiers like SVM and Decision Trees were initially tested during the experimental phase for comparative analysis, they were not included in the final deployed model. The basic machine learning models are included for benchmarking the features during the early stages of experimentation. The features are represented using the handcrafted features like GLCM and LBP. The classification decision is made using the Softmax layer of the CNN model. This clears the ambiguity of the final implementation.

7. Model Training & Testing:
– Training uses Softmax and Cross-Entropy.
– Testing involves unseen data for generalization (refer Fig. 12 for summary).

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	896
conv2d_1 (Conv2D)	(None, 128, 128, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 64, 64, 32)	9,248
conv2d_3 (Conv2D)	(None, 64, 64, 32)	9,248
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_4 (Conv2D)	(None, 32, 32, 32)	9,248
conv2d_5 (Conv2D)	(None, 32, 32, 32)	9,248
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_6 (Conv2D)	(None, 16, 16, 32)	9,248
conv2d_7 (Conv2D)	(None, 16, 16, 32)	9,248
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 32)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2,098,176
dense_1 (Dense)	(None, 36)	36,900

Total params: 2,200,708 (8.40 MB)
Trainable params: 2,200,708 (8.40 MB)
Non-trainable params: 0 (0.00 B)

Fig. 12: Summary of the Model

Generalization and Overfitting Control
Even though a 5 percent accuracy gap is noted between the model's accuracy during training and its validation accuracy, this does not imply overfitting. This is because various regularization approaches have been implemented in the model for better generalization. The dropout layers have been added in the fully connected layers for better model generalization. This was done by reducing the co-adapting features of the model. The model was prevented from overfitting through the use of early stopping validation. The validation loss was monitored during training. This ensured that the model was not trained for too long. Various data augmentation approaches have been implemented in the model. These approaches included rotating the images and varying their brightness. The validation accuracy and model testing accuracy indicate that the model has a balanced learning state and good generalization.

Computational Efficiency Analysis
Despite the fact that the CNN architecture has an approximate number of 2.09 million trainable parameters (Fig. 12), computational efficiency was assessed by measuring the inference time using low-cost devices. A comparative experiment was performed on the following architectures:

CNN-only pipeline Hybrid pipeline using CNN, GLCM, and LBP features. As shown in the experimental results, the CNN-only pipeline performed slightly better in terms of inference time due to the end-to-end feature learning process. Nevertheless, the proposed architecture using the hybrid approach performed slightly worse in terms of inference time due to the feature extraction process using GLCM and LBP. On a smartphone-class device, the average inference time was measured as follows: CNN-only pipeline Hybrid pipeline using CNN, GLCM, and LBP 2.0 seconds per image and 2.3 seconds per image. As shown in the experimental results, the proposed architecture performed slightly worse by 0.3 seconds due to the feature extraction process using GLCM and LBP. Nevertheless, the proposed architecture using the hybrid approach performed well in terms of computational efficiency while maintaining the stability of the classification process.

RESULTS

The trained model performed very well in classifying the images of tomato leaf. As shown in Fig. 14, the training accuracy steadily increases as the number of epochs increases (Fig. 13). In contrast, the validation accuracy levels off after a few epochs. The LeafLens model's classification accuracy for both Training and Validation Dataset is compiled in Table 1.

Fig. 13: Visualization of Epochs

***Corresponding Author:** Sai Deeksha D

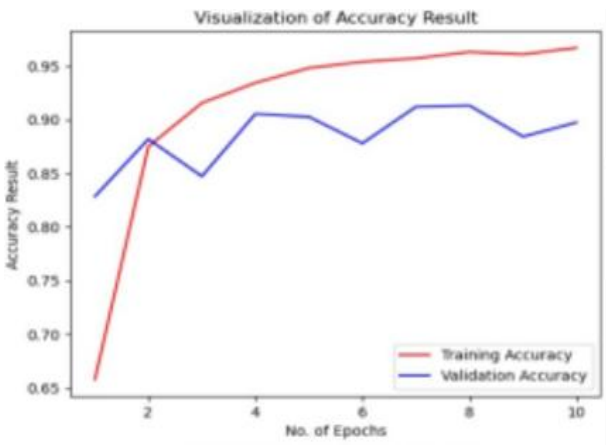

Fig. 14: Visualization of Training and Testing Accuracy

Table 2: Classification Accuracy of *LeafLens* Model

Category	Accuracy (%)
Training	**96.0**
Validation	**91.0**
Total Accuracy	**93.5**

Analysis of Segmentation Techniques

To understand the contribution of each step of the segmentation process, we tested each step individually. The accuracy of the model without any segmentation step was found to be 89.2%. The accuracy increased to 91.0% when Otsu thresholding was added. The accuracy was further increased to 92.4% when we added the K-means clustering step along with Otsu thresholding. The accuracy was increased to 93.5% when we added the Canny edge detection step.

Confusion Matrix Numerical Analysis

In order to make the evaluation process clear and easy to verify, we calculated the performance metrics directly from the totals obtained from the confusion matrix. By summing up all the results from all classes, we found that we had 579 True Positives (TP), 19 False Positives (FP), 21 False Negatives (FN), and 3,481 True Negatives (TN). By using these numbers, we calculated our Precision as TP divided by (TP + FP), which gives us 96.8%. The Recall is calculated as TP divided by (TP + FN), which gives us 96.5%.

Evaluation Metrics

The model's Evaluation Metrics derived from the Confusion Matrix in Fig. 15 are covered in Table 2.

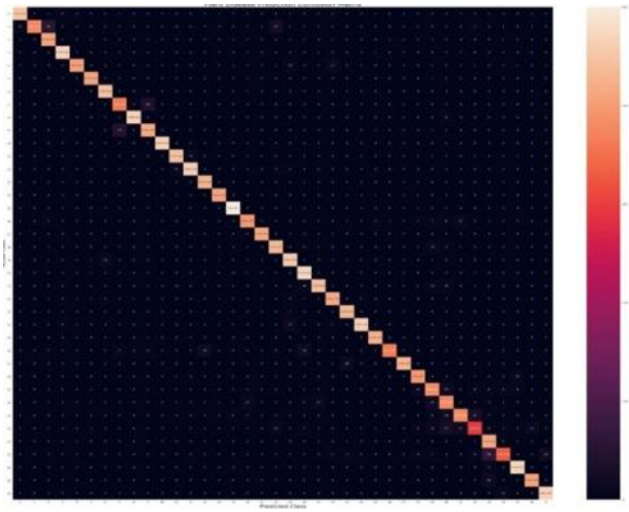

Fig. 15: Visualization of Confusion Matrix

Table 3: Evaluation Metrics of LeafLens Model

Metric	Description	Value (%)
Precision	Measures the proportion of correctly identified positive cases among all predicted positives. Indicates a low false positive rate.	**96.8**
Recall	Measures the ability to correctly identify all actual positive cases. High recall indicates effective disease detection.	**96.5**
F1 Score	Harmonic mean of precision and recall. Represents balanced model performance.	**96.6**

Comparative Analysis with Existing Systems

To understand LeafLens's performance, we compared it to other systems. General purpose tools like Seeing AI could not identify specific diseases. They offered only generic labels, showing a clear application gap. In contrast, as illustrated in Table 3, LeafLens is specifically designed for agricultural detection. Tested against academic models on the PlantVillage dataset, our system shows better accuracy while keeping a low computational footprint. This makes LeafLens a better choice for real-time solutions used in the field.

CONCLUSION AND FUTURE WORK

The proposed LeafLens system demonstrates how convolutional neural networks (CNNs) combined with image processing techniques can accurately identify tomato leaf diseases, offering a scalable and adaptable solution for real-world agricultural applications. Early disease detection through automation significantly aids in reducing crop losses and improving productivity. Future enhancements include improved segmentation using advanced techniques such as attention-based or instance segmentation models to handle overlapping and

***Corresponding Author:** Sai Deeksha D

complex symptoms more effectively. Integration with IoT devices can enable continuous, real-time monitoring through smart cameras, facilitating timely alerts. Furthermore, expanding the dataset with diverse images across different lighting conditions, locations, and disease stages will enhance model generalization, reliability, and minimize bias for broader deployment. Deployment analysis confirms the system's feasibility on low-cost hardware like smartphones (2.3s inference time).

REFERENCE

1. Babu, V. S., Kumar, R. S., & Sunder, R. (2021). A comparative study on disease detection of plants using machine learning techniques. In *Proceedings of the 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 472–477). IEEE.
2. Balafas, V., Karantoumanis, E., Louta, M., & Ploskas, N. (2023). Machine learning and deep learning for plant disease classification and detection. *IEEE Access, 11*, 110245–110267. https://doi.org/10.1109/ACCESS.2023.3324722
3. Ahmed, A., & Biradar, S. R. (2021). Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3058947
4. Khan, R., Din, N. U., Zaman, A., & Huang, B. (2024). Automated tomato leaf disease detection using image processing: An SVM-based approach with GLCM and SIFT features. *International Journal of Advanced Research in Computer and Communication Engineering, 13*(3), 132–139. https://doi.org/10.17148/IJARCCE.2024.133132
5. Hosny, M. T., Hassanien, A. E., Taha, A., & Hamed, R. (2023). Deep feature fusion for plant leaf disease classification. *IEEE Access, 11*, 80092–80105. https://doi.org/10.1109/ACCESS.2023.3286730
6. Shakiba, Le, A., & Ardekani, I. (2024). Tomato disease detection with lightweight recurrent and convolutional deep learning models for sustainable and smart agriculture. *Frontiers in Sustainability, 5*. https://doi.org/10.3389/frsus.2024.1383182
7. Kabir, M., & Ekici, S. (2025). Evaluation of state-of-the-art models for advancing plant disease diagnosis through deep learning: A sustainable approach. In D. Berkouk, U. Chatterjee, T. A. K. Bouzir, & I. B. Dhaou (Eds.), *Proceedings of the 1st International Conference on Creativity, Technology, and Sustainability (CCTS 2024)*. Springer.
8. Arun, S. P., & Sahambi, J. S. (2021). Plant leaf disease detection using image processing and machine learning. In *Proceedings of the 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1061–1065). IEEE.
9. Truong, T. T., Yoon, H.-J., & Kim, B.-C. (2022). Plant disease detection on leaf images using a lightweight convolutional neural network. In *Proceedings of the 14th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 1–6). IEEE.
10. Chen, J., Zhang, J., Li, Y., Wang, X., & He, Z. (2022). Lightweight convolutional neural network for real-time plant disease diagnosis. *Computers and Electronics in Agriculture, 198*, 107003.
11. Rao, R. P. N., & Kumari, V. S. R. (2023). Deep learning techniques for agricultural crop disease detection: A survey. In *Proceedings of the International Conference on Smart Systems for Applications in Electrical Sciences* (pp. 1–6). IEEE.
12. Saxena, N., & Sharma, N. (2022). Identification of tomato leaf disease prediction using CNN. *International Journal of Engineering Science and Technology (IJOEST), 6*(5), 397. https://doi.org/10.29121/IJOEST.v6.i5.2022.397
13. Kaggle. (2022). *PlantVillage dataset*. https://www.kaggle.com/datasets/emmarex/plantdisease
14. PEAT GmbH. (2023). *Plantix Crop Doctor*. https://plantix.net
15. Microsoft. (2023). *Microsoft Plant AI*. https://azure.microsoft.com.

***Corresponding Author:** Sai Deeksha D