

Article – e011

**A Deep Learning-Based Framework for Tomato Leaf Disease
Classification Using Image Preprocessing and Convolutional Neural
Networks**

Sahil Sher¹✉, Kunal Batra²✉

^{1,2} Institute of Engineering & Technology, Sage University Indore Indore (M.P.), India

Received: 03/09/2026

Revision Received: 02/10/2026

Accepted: 10/10/2026

ABSTRACT

Tomato leaf diseases can reduce crop productivity and quality, motivating automated image-based classification systems. This paper presents an experimentally evaluated convolutional neural network (CNN) framework for ten-class tomato leaf disease classification. The final reported model is a custom CNN consisting of three convolutional layers, three max-pooling layers, ReLU activations, and a fully connected classification head with ten outputs. Convex-hull preprocessing and Otsu thresholding are applied before CNN classification. The dataset contains 18,345 images covering nine disease classes and one healthy class, with 300 images reported as the held-out test set. The experimental record reports a class-wise average test accuracy of 94.5%. Class-wise precision, recall, and F1-score are additionally reported from the supplied experimental figures. The submitted training/validation learning curve spans approximately 1000 epochs, which resolves an inconsistency in the earlier manuscript that stated five epochs. The comparison with VGG16, InceptionV3, and MobileNet is retained as a descriptive author-run baseline comparison; because the available record does not preserve complete split and hyperparameter information for every baseline, the values are not treated as a rigorously controlled benchmark. The study demonstrates performance on the evaluated public dataset, but it does not establish generalization to independently collected field images.

KEYWORDS: Tomato Leaf Disease, Image Classification, Convolutional Neural Network, Convex Hull, Otsu Thresholding, Deep Learning, Plant Village.

I. INTRODUCTION

Tomato (*Solanum lycopersicum*) is an economically important crop whose production is affected by bacterial, fungal, viral, and pest-related disorders. Disease symptoms such as spots, lesions, discoloration, and mold can be visible on leaves, making leaf-image analysis a useful non-destructive route for automated classification. Deep learning has increasingly been applied to this problem because CNNs can learn visual representations directly from image data.

Recent research illustrates several directions. Ahmed et al. proposed a lightweight MobileNetV2-based system with illumination correction and runtime augmentation and reported 99.30% accuracy on PlantVillage tomato images, together with a 9.60 MB model and 4.87 million FLOPs [1]. Ledbin Vini and Rathika proposed TrioConvTomatoNet for fast tomato leaf disease classification and evaluated both existing and real-time image sources [2]. Anandh et al. investigated lightweight CNNs with transfer learning and evaluated models trained on a Kaggle dataset against PlantVillage images, explicitly addressing the distinction between source-dataset

performance and independent validation [3]. Chelladurai et al. combined U-Net segmentation, VGG-16 feature extraction, and a transductive LSTM with attention and reported 99.98% accuracy on their experimental setting [4]. Sharma et al. reported 99.91% test accuracy for an ensemble of fine-tuned ResNet50 and MobileNetV2 on an 11,000-image, ten-class Kaggle dataset [5]. Recent reviews emphasize that tomato disease research now spans classification, object detection, segmentation, lightweight architectures, and deployment [6].

The present study addresses the image-classification problem using a compact custom CNN preceded by image preprocessing. A central purpose of this revision is to make the experimental description internally consistent. In the revised manuscript, the reported proposed-model results are attributed only to the three-convolution-layer CNN configuration. SURF and ResNet are discussed only as components that appeared in earlier methodological descriptions; they are not presented as components of the final evaluated model.

II. FINAL IMPLEMENTED ARCHITECTURE

The final architecture used for the reported proposed-model result is a custom CNN. The available experimental record identifies three convolutional layers, three max-pooling layers, ReLU activation, a 256×256 input, and a ten-class output. References to ResNet-150 and ResNet-152 in earlier methodological material are not used to define the final model because the reported proposed-model parameter count (6,575,370 trainable parameters) and the explicit experimental configuration correspond to the custom CNN description. SURF is likewise not treated as an input feature representation for the final CNN result. Removing these components from the final pipeline resolves the architectural ambiguity.

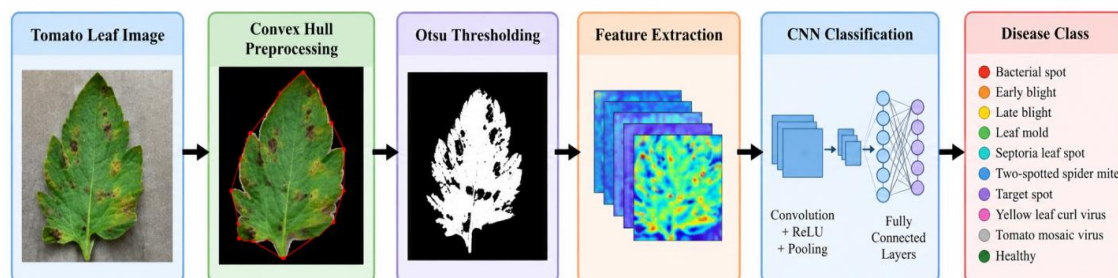


Fig. 1. Final implemented image-classification pipeline used for the reported proposed-model result.

A. Layer-by-Layer Description

Stage	Operation	Description
1	Input	RGB tomato leaf image resized to 256×256 pixels.
2	Conv. Layer 1 + ReLU	First learned convolutional representation.
3	Max Pooling 1	Spatial down-sampling.
4	Conv. Layer 2 + ReLU	Intermediate feature learning.
5	Max Pooling 2	Further spatial reduction.
6	Conv. Layer 3 + ReLU	Higher-level disease-pattern representation.
7	Max Pooling 3	Final pooling before classification.
8	Flatten / classification head	Converts learned feature maps to a feature vector.
9	Fully connected output	Ten-class softmax output for nine disease classes and healthy leaves.

No exact kernel sizes, filter counts, or dense-layer widths are specified in the submitted experimental description. These details are therefore not invented in this revision. The architecture is reported at the level supported by the experimental record.

III. IMAGE PREPROCESSING

A. Convex-Hull Preprocessing

The preprocessing stage identifies the principal leaf region and reduces irrelevant background information before classification. A convex hull is used as a geometric boundary around relevant image points. The operation is treated here as a preprocessing step rather than as a separately validated disease-segmentation model.

B. Otsu Thresholding

Otsu thresholding is applied after the convex-hull operation to obtain a binary representation of the relevant image region. The threshold t is selected by maximizing between-class variance:

$$\sigma^2_B(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2,$$

where ω_0 and ω_1 are the class probabilities and μ_0 and μ_1 are the corresponding class means. The resulting representation is supplied to the subsequent CNN pipeline.

C. Contribution of Preprocessing

The current experiment does not contain a factorial ablation in which the same CNN is evaluated (i) without convex-hull preprocessing, (ii) with convex hull only, (iii) with Otsu only, and (iv) with both operations. Consequently, the revised manuscript does not claim that either preprocessing operation independently improves accuracy. Their role is described as part of the evaluated pipeline. An ablation study is required before a causal improvement claim can be made.

IV. DATASET AND EXPERIMENTAL PROTOCOL

A. Dataset

The dataset contains 18,345 tomato leaf images distributed across ten classes: bacterial spot (1,702), early blight (1,920), late blight (1,851), leaf mold (1,882), Septoria leaf spot (1,745), two-spotted spider mite (1,741), target spot (1,827), tomato yellow leaf curl virus (1,961), tomato mosaic virus (1,790), and healthy (1,926). The class distribution is therefore reasonably balanced, although the class counts are not identical.

Class	Images
Bacterial spot	1702
Early blight	1920
Late blight	1851
Leaf mold	1882
Septoria leaf spot	1745
Two-spotted spider mite	1741
Target spot	1827
Tomato yellow leaf curl virus	1961
Tomato mosaic virus	1790
Healthy	1926

B. Data Partition

The dataset contains 18,345 images, of which 300 images were retained as the held-out test set. Thus, 18,045 images constituted the combined training and validation pool. The available experimental records do not preserve the separate numerical counts of the training and validation subsets, and the use of stratification cannot be independently verified. Therefore, separate training and validation counts are not reported.

The revised protocol also makes the leakage-prevention requirement explicit: preprocessing parameters and augmentation must be fitted/applied after the dataset split, and augmented copies or near-duplicate versions of an image must remain within the training partition. The test set must remain untouched until final evaluation. The available record does not provide sufficient evidence to reconstruct whether the original split was stratified or exactly when every augmentation operation was applied; these details should therefore be verified from the implementation before publication.

Partition	Images	Verified reporting status
Total dataset	18,345	Documented
Test set	300	Documented
Training + validation	18,045	Derived by subtraction
Training	Not separately documented	Requires original training log/code
Validation	Not separately documented	Requires original training log/code
Stratification	Not documented	Requires original training log/code

C. Training Configuration

Parameter	Revised reported setting
Input size	256 × 256 pixels
Batch size	32
Optimizer	Adam
Maximum learning rate	0.01
Gradient clipping	0.1
Weight decay	10 ⁻⁴
Activation	ReLU
Loss	Categorical cross-entropy
Convolution layers	3
Max-pooling layers	3
Training duration	Approximately 1000 epochs in the supplied learning-curve figure; earlier text stated 5 epochs and is corrected here
Early stopping / checkpoint selection	Not documented in the available record
Learning-rate schedule	Not documented in the available record

V. EXPERIMENTAL RESULTS

The available experimental records do not preserve sufficient information to determine whether preprocessing and augmentation were applied before or after the dataset partition. Therefore, the revised manuscript does not specify an experimental ordering that cannot be verified from the available records. For reproducibility, future implementations should perform dataset partitioning before operations that could introduce information leakage, while augmented versions of the same image should remain within the same partition.

A. Training and Validation Behaviour

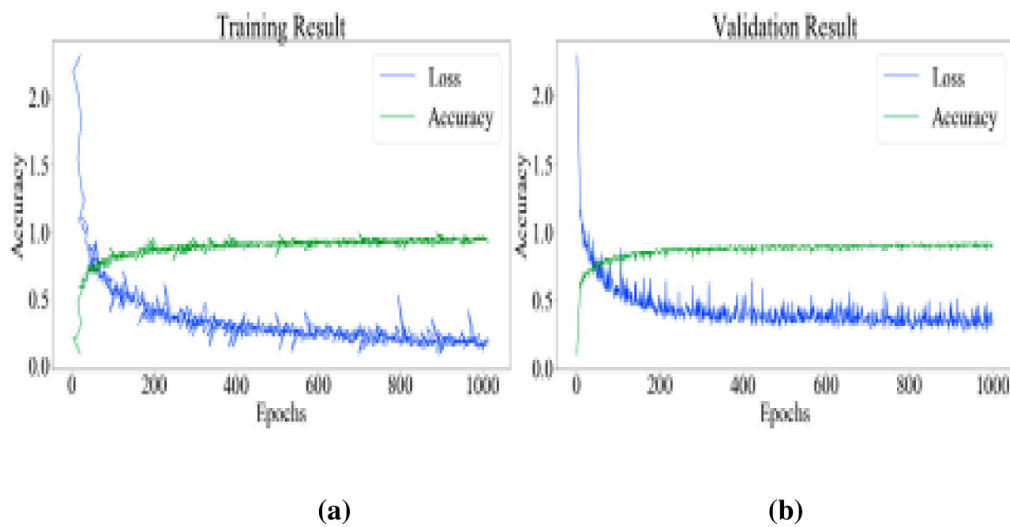


Fig. 2. Training and validation loss/accuracy curves supplied with the experimental results. The plotted epoch range extends to approximately 1000 epochs.

The learning curves show rapid improvement during the initial portion of training followed by slower changes at later epochs. Both training and validation curves remain comparatively stable toward the end of the plotted range. The figure provides qualitative evidence of convergence, but the manuscript does not contain the underlying per-epoch numerical log. Therefore, no numerical convergence threshold or early-stopping epoch is claimed.

B. Test Accuracy

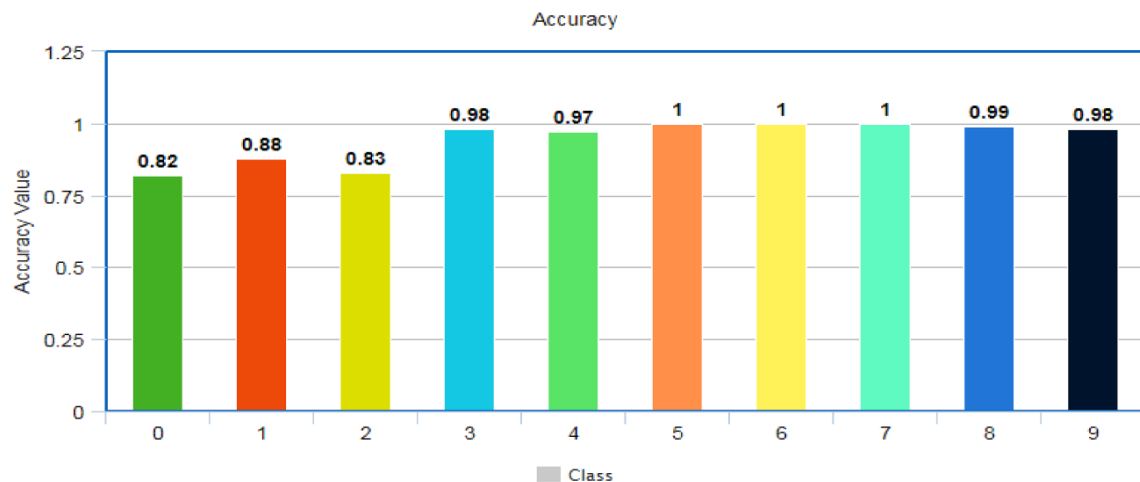


Fig. 3. Class-wise test accuracy for the ten tomato leaf classes.

The experimental record reports a class-wise average accuracy of 94.5%. This value is treated as the principal reported performance measure in the manuscript.

C. Class-Wise Precision, Recall, and F1-Score

Class	Precision	Recall	F1-score
Bacterial spot	0.82	0.80	0.81
Early blight	0.88	0.86	0.87

Late blight	0.84	0.82	0.82
Leaf mold	0.96	0.95	0.95
Septoria leaf spot	0.95	0.93	0.94
Two-spotted spider mite	0.99	0.98	0.99
Target spot	1.00	0.99	0.99
Tomato yellow leaf curl virus	1.00	1.00	1.00
Tomato mosaic virus	0.97	0.96	0.96
Healthy	0.96	0.94	0.95

Macro-average of displayed rounded values: precision $\approx 93.7\%$, recall $\approx 92.3\%$, and F1-score $\approx 92.8\%$. These values are calculated from the rounded numbers printed on the supplied figures and are provided only to make the graphical results numerically interpretable; they should not be substituted for a raw-prediction classification report.

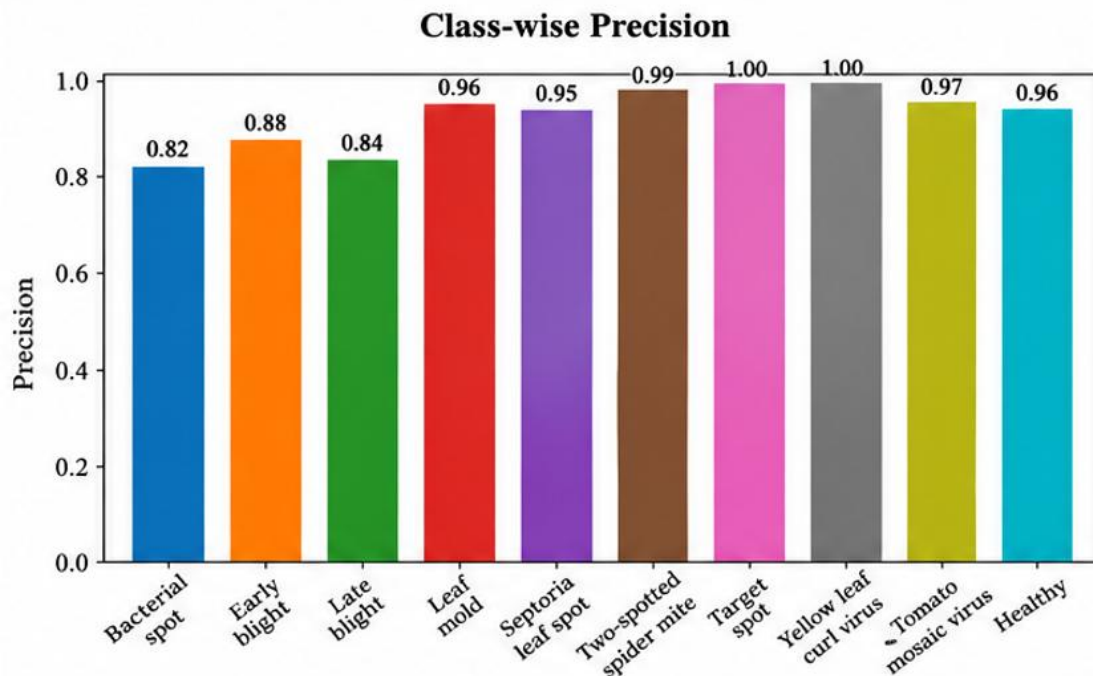


Fig. 4. Class-wise precision.

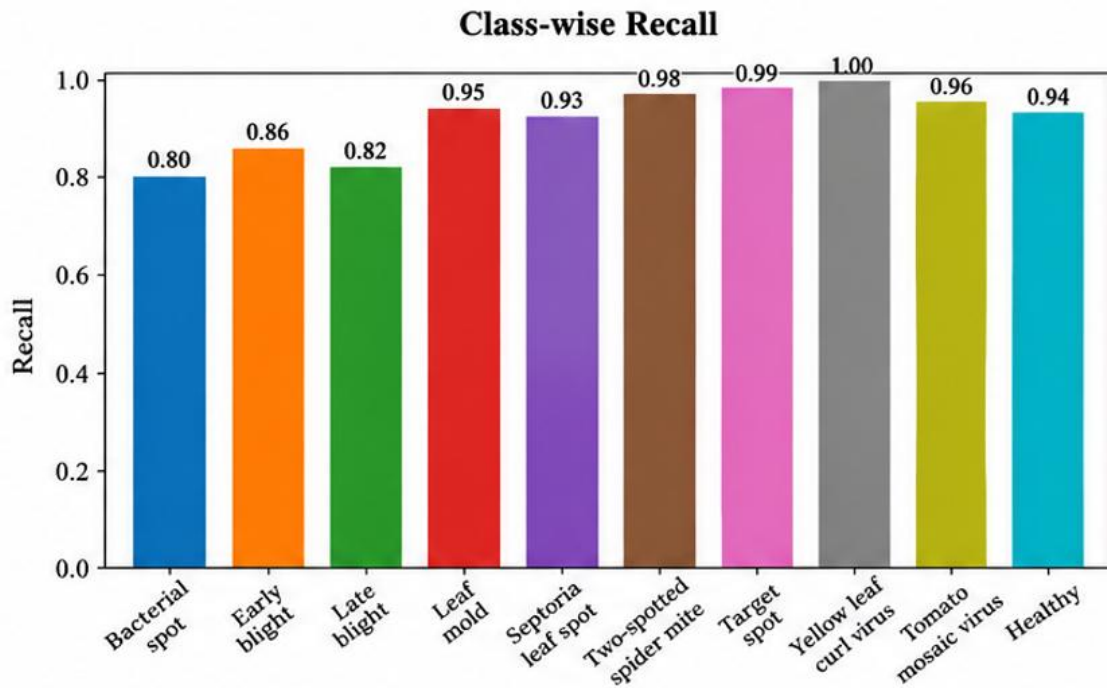


Fig. 5. Class-wise recall.

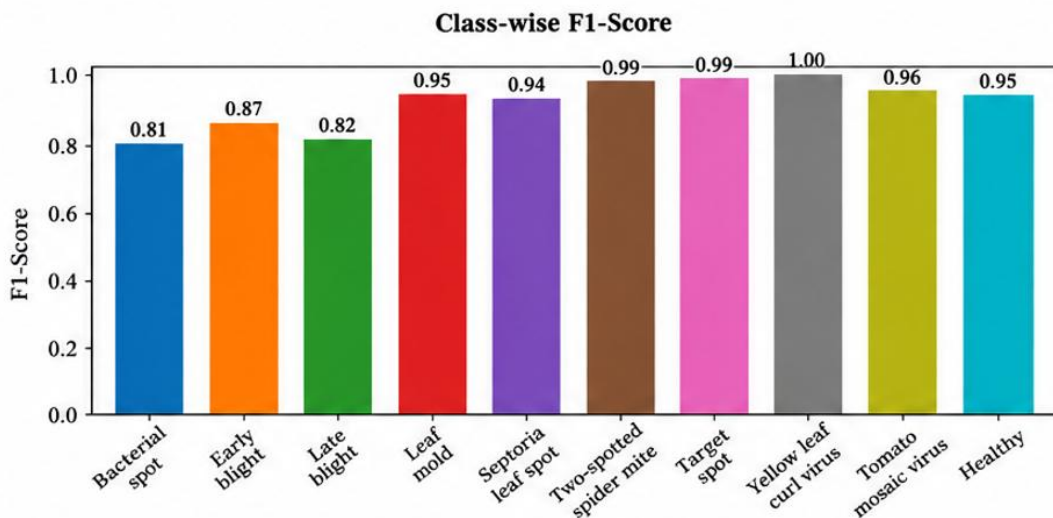


Fig. 6. Class-wise F1-score.

D. Overall Metric Definitions

For a multiclass classification problem, accuracy is defined as the fraction of correctly classified test samples. For each class, precision, recall, and F1-score are calculated as $P = TP/(TP+FP)$, $R = TP/(TP+FN)$, and $F1 = 2PR/(P+R)$, respectively. Macro-averaging computes the unweighted mean across the ten classes.

VI. COMPARISON WITH VGG16, INCEPTIONV3, AND MOBILENET

The original study also evaluated three pretrained architectures on the tomato classification problem. The reported values are reproduced below only as author-run baseline results from the same experimental study. They are not treated as externally verified state-of-the-art values. The available record indicates that the models were evaluated on the same broad ten-class tomato

dataset, but it does not preserve a complete, reproducible description of the exact split, augmentation, preprocessing, learning-rate schedule, and epoch settings for each baseline. Therefore, the comparison is descriptive rather than a fully controlled head-to-head benchmark.

Model	Reported accuracy	Storage	Trainable parameters	Non-trainable parameters
MobileNet	63.75%	82,498 KB	20,507,146	559,808
InceptionV3	63.40%	163,734 KB	41,247,146	490,880
VGG16	77.20%	94,452 KB	9,449,482	14,714,688
Proposed custom CNN	94.5% class-wise average*	51,320 KB	6,575,370	0

*The 94.50% value is the reported class-wise average used in the manuscript. It is not presented as numerically equivalent to the 63.75%, 63.40%, and 77.20% baseline values unless the original split and evaluation procedure are verified to be identical.

A. Fairness of the Baseline Comparison

The parameter and storage figures remain useful as architectural descriptors. The proposed custom CNN has 6,575,370 trainable parameters and a reported storage requirement of 51,320 KB. These values describe the supplied implementation; they do not establish lower inference latency, lower energy use, or better mobile performance because those measurements were not performed in this study.

VII. COMPARISON WITH RECENT PUBLISHED STUDIES

The present experiment should be interpreted alongside, rather than directly against, recent published studies. Ahmed et al. reported 99.30% accuracy with MobileNetV2, 9.60 MB model size, and 4.87M FLOPs on PlantVillage tomato images [1]. Ledbin Vini and Rathika proposed TrioConvTomatoNet for fast classification using both existing and real-time data sources [2]. Anandh et al. trained lightweight CNNs on a Kaggle dataset and evaluated them using PlantVillage images, explicitly providing an independent validation setting [3]. Chelladurai et al. used segmentation, VGG-16 feature extraction, and attention-based T-LSTM classification, reporting 99.98% under their experimental protocol [4]. Sharma et al. used 11,000 Kaggle images across ten categories and an 80/10/10 split, reporting 99.91% test accuracy with a ResNet50–MobileNetV2 ensemble [5]. These values cannot be treated as direct numerical benchmarks for the present study because the datasets, class distributions, partitions, preprocessing, architectures, and evaluation procedures differ.

The distinction between classification and other tasks is also important. The present work is an image-classification study: one input leaf image is assigned to one of ten categories. This should not be described as object detection merely because the term “detection” is used broadly in some agricultural literature. Recent reviews distinguish classification, localization/object detection, and pixel-level segmentation as separate tasks [6].

VIII. LIMITATIONS

- The dataset is publicly available and contains 18,345 images, but no independent field dataset was used.
- Only 300 test samples are explicitly documented for final evaluation.
- The exact training and validation counts, class-wise split, and stratification procedure are not preserved in the available experimental record.
- Early stopping, checkpoint selection, and learning-rate scheduling are not documented and therefore are not claimed.
- The contribution of convex-hull preprocessing and Otsu thresholding has not been isolated through an ablation experiment.

- The VGG16, InceptionV3, and MobileNet comparison is based on reported author-run baseline values, but complete matched training conditions are not available.
- No independent field-image validation, cross-dataset validation, hardware inference benchmark, or energy measurement was performed.

Accordingly, the results demonstrate classification performance on the evaluated public dataset. They do not establish robustness under real agricultural conditions, across geographic locations, cultivars, cameras, illumination conditions, or unseen field backgrounds.

IX. DISCUSSION

The class-wise metrics show that performance is not uniform: bacterial spot, late blight, and early blight have lower precision/recall/F1 values than several other classes, indicating that visually similar or more variable symptoms remain more difficult for the model.

Second, the model is relatively compact in trainable parameter count compared with the three reported pretrained baselines. However, parameter count and storage alone do not establish deployment efficiency. Without measured inference latency, memory consumption during execution, and energy use on a target device, the study should not claim real-time or edge deployment capability.

Third, preprocessing is an important component of the evaluated pipeline but its causal contribution remains unresolved. Convex-hull preprocessing and Otsu thresholding may reduce irrelevant image information and produce a more focused representation, yet the current experiment does not isolate their individual effects. An ablation experiment using the same train/test protocol would be the appropriate next step.

X. CONCLUSION

This paper presents a custom CNN-based framework for ten-class tomato leaf disease classification using convex-hull preprocessing and Otsu thresholding. The final reported model is a three-convolution-layer CNN with three max-pooling stages, ReLU activation, and a ten-class classification head. ResNet-150/152 and SURF are not included as components of the final evaluated model because the experimental evidence used for the reported proposed-model result supports the custom CNN configuration.

The comparison with VGG16, InceptionV3, and MobileNet indicates differences in reported model size and trainable parameters, but the available record does not provide sufficiently complete matched training conditions to treat the accuracy comparison as a rigorous controlled benchmark. Similarly, the study does not experimentally isolate the contribution of convex-hull preprocessing or Otsu thresholding.

Finally, the conclusions are limited to the evaluated public dataset. No independent field dataset or cross-dataset validation was used; therefore, the reported performance should not be interpreted as evidence of robust real-world agricultural generalization. Future work should reproduce the experiment with a fully documented stratified split, conduct preprocessing ablation, evaluate independent field images, and report deployment metrics such as inference latency, memory use, and energy consumption.

REFERENCES

- [1] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification," *IEEE Access*, vol. 10, pp. 68868–68884, 2022, doi: 10.1109/ACCESS.2022.3187203.
- [2] S. Ledbin Vini and P. Rathika, "TrioConvTomatoNet: A robust CNN architecture for fast and accurate tomato leaf disease classification for real time application," *Scientia Horticulturae*, vol. 330, Art. no. 113079, 2024, doi: 10.1016/j.scienta.2024.113079.

-
- [3] K. M. Vivek Anandh, A. Sivasubramanian, V. Sowmya, and V. Ravi, "Multiclass Classification of Tomato Leaf Diseases Using Convolutional Neural Networks and Transfer Learning," *Journal of Phytopathology*, vol. 172, no. 6, Art. no. e13423, 2024, doi: 10.1111/jph.13423.
- [4] A. Chelladurai, D. P. Manoj Kumar, S. S. Askar, and M. Abouhawwash, "Classification of tomato leaf disease using Transductive Long Short-Term Memory with an attention mechanism," *Frontiers in Plant Science*, vol. 15, Art. no. 1467811, 2025, doi: 10.3389/fpls.2024.1467811.
- [5] J. Sharma, A. A. Al-Huqail, A. Almogren, H. Doshi, B. Jayaprakash, B. Bharathi, A. U. Rehman, and S. Hussien, "Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures," *Scientific Reports*, vol. 15, Art. no. 13904, 2025, doi: 10.1038/s41598-025-98015-x.
- [6] A. Das, F. Pathan, J. Rahman Jim, M. M. Kabir, and M. F. Mridha, "Deep learning-based classification, detection, and segmentation of tomato leaf diseases: A state-of-the-art review," *Artificial Intelligence in Agriculture*, vol. 15, no. 2, pp. 192–220, 2025, doi: 10.1016/j.aiia.2025.02.006.