Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Classical Machine Learning versus Transfer Learning for Tomato Leaf Disease Classification: A Reproducible Comparative Framework

Jorge manuel Barrios Sánchez
jm.barriossanchez@ugto.mx

Institución Universitaria Mayor de Cartagena
Fredy Alberto Munera Romero
Corporación Universitaria Rafael Nuñez
Jose Eleazar Peralta Lopez
Instituto Tecnológico Superior de Salvatierra
Jose Manuel Lopez Villagomez
Instituto Tecnológico Superior de Salvatierra
Ernestina Becerra Becerra
Instituto Tecnológico Superior de Salvatierra
A. Nadin Lule-Chávez
Instituto Tecnológico Superior de Salvatierra
Ana Laura Martínez
Universidad de Guanajuato
Adrián Felipe Gómez Consuegra
Corporación Universitaria Rafael Nuñez

Research Article

Keywords: machine learning, tomato leaf disease, computer vision, PlantVillage, transfer learning, MobileNetV2, support vector machine, Random Forest, smart agriculture

Posted Date: September 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10830175/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Classical Machine Learning versus Transfer Learning for Tomato Leaf Disease Classification: A Reproducible Comparative Framework

Jorge Manuel Barrios Sánchez[1,2*], Fredy Alberto Múnera Romero[2], José Eleazar Peralta López[3], José Manuel López Villagómez[3,4], Ernestina Becerra Becerra[3], A. Nadin Lule Chávez[3], Ana Laura Martínez Herrera[5*], Adrián Felipe Gómez Consuegra[2]

[1]Institución Universitaria Mayor de Cartagena, Cartagena de Indias, Colombia .
[2]Programa de Ingeniería de Sistemas, Facultad de Ingeniería, Corporación Universitaria Rafael Núñez, Cartagena de Indias, Colombia .
[3]Instituto Tecnológico Superior de Salvatierra, Salvatierra, Guanajuato, Mexico .
[4]Universidad Virtual del Estado de Guanajuato, Guanajuato, Mexico .
[5]Universidad de Guanajuato, Guanajuato, Mexico .

*Corresponding author(s). E-mail(s): jm.barriossanchez@ugto.mx; martinez.al@ugto.mx;
Contributing authors: fmunerar21@campusuninunez.edu.co; joperalta@itess.edu.mx; jolopezv@uveg.edu.mx; erbecerra@itess.edu.co; anule@itess.edu.mx; agomezc21@campusuninunez.edu.co;

Abstract

Automated plant-disease recognition is increasingly used as a benchmark application for computer vision in precision agriculture, yet the relative benefit of classical machine learning and transfer learning depends on feature representation and experimental design. This study presents a reproducible comparison of K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and MobileNetV2 for ten-class tomato leaf classification using the PlantVillage dataset. The 18,178 original images were partitioned class-wise into training

(12,720), validation (2,722), and independent test (2,736) subsets. Only the training subset was balanced through image augmentation, producing 37,490 training images. Classical models used a 110-dimensional descriptor combining HSV color histograms, gray-level co-occurrence matrix statistics, and local binary patterns, whereas MobileNetV2 used ImageNet transfer learning with model-specific preprocessing and two-stage fine-tuning. On the independent test set, MobileNetV2 achieved the highest numerical accuracy (97.26%), SVM-RBF achieved 97.11% accuracy and the highest macro-F1 score (96.47%), and RF achieved 96.93% accuracy and the highest macro-recall (96.69%). KNN achieved 58.22% accuracy. Wilson confidence intervals and stratified bootstrap analysis showed closely overlapping uncertainty ranges for SVM-RBF, RF, and MobileNetV2. Exact paired McNemar tests with Holm correction found no statistically significant differences in classification correctness among these three high-performing models, whereas each significantly outperformed KNN. The results indicate that compact handcrafted color-texture representations combined with nonlinear classical classifiers can achieve performance comparable to transfer learning under controlled PlantVillage conditions. External validation with field-acquired images remains necessary before inferring real-world diagnostic performance.

Keywords: machine learning, tomato leaf disease, computer vision, PlantVillage, transfer learning, MobileNetV2, support vector machine, Random Forest, smart agriculture

1 Introduction

Agricultural production is highly vulnerable to pests and plant diseases, which can reduce crop yield, increase production costs, and affect food availability. In horticultural crops such as tomato (*Solanum lycopersicum*), foliar diseases are particularly relevant because many symptoms initially appear on the leaves. Early recognition of these symptoms can support timely phytosanitary decisions, reduce inadequate pesticide application, and minimize losses associated with delayed or inaccurate diagnosis. However, in many rural and resource-constrained agricultural contexts, disease identification still relies on visual inspection by farmers or on the availability of specialized technical assistance, which may be limited, costly, or delayed.

Tomato has become one of the most widely investigated crops in automatic plant disease recognition because of its economic relevance and the visual diversity of its foliar diseases. Bacterial spot, early blight, late blight, leaf mold, Septoria leaf spot, target spot, tomato mosaic virus, tomato yellow leaf curl virus, and spider mite damage are among the categories commonly represented in public image datasets. PlantVillage has become one of the most frequently used benchmark datasets for plant disease recognition because it provides a large collection of labeled images acquired under relatively controlled conditions [1]. Although these characteristics facilitate reproducible experimentation, they also mean that results obtained from PlantVillage should not be interpreted directly as evidence of performance under real field conditions.

Classical image processing and machine learning approaches remain relevant for plant disease classification. These methods typically extract color, shape, or texture

characteristics from leaf images and use the resulting numerical representations as input to classifiers such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and related algorithms [2–4]. Their relatively low computational requirements and explicit feature representations make them particularly useful for comparative studies and resource-constrained applications. Nevertheless, their performance depends strongly on the discriminative capacity of the selected features and on the variability of image acquisition conditions.

Convolutional neural networks (CNNs), in contrast, learn hierarchical representations directly from image pixels and have demonstrated strong performance in plant disease recognition. Several investigations have applied CNN-based architectures to tomato leaves and broader plant disease datasets, reporting high classification performance under controlled experimental conditions [5–8]. Deep learning reduces the need for manually designed feature representations but generally involves greater computational requirements and additional design choices related to network architecture, training strategy, regularization, and data augmentation.

Other studies have extended plant disease recognition beyond direct classification by incorporating segmentation, localization, and more comprehensive visual analysis strategies. Segmentation approaches can isolate relevant leaf regions before classification [9], while integrated localization and classification frameworks have been proposed to identify disease symptoms more robustly [10]. These developments demonstrate that plant disease recognition is influenced not only by classification algorithms but also by background characteristics, lesion distribution, symptom intensity, illumination, and image-acquisition conditions.

A particularly relevant research question is whether the additional complexity of deep learning necessarily results in superior predictive performance when images are acquired under controlled conditions. Tan et al. [11] compared classical machine learning and deep learning methods for tomato leaf disease classification and highlighted the importance of feature representation and common experimental conditions. Reviews of plant disease recognition have similarly emphasized that reported performance can be strongly influenced by preprocessing procedures, dataset composition, disease severity, feature extraction, and validation strategies [12]. Consequently, meaningful model comparisons require consistent dataset partitions, preprocessing protocols, and evaluation metrics.

Transfer learning has become an important strategy for reducing the computational and data requirements associated with training deep neural networks from scratch. Previous studies have investigated feature concatenation and transfer learning for tomato disease classification [13], transfer learning-based deep convolutional architectures [14], and frameworks for tomato pest and disease recognition [15]. Additional approaches have incorporated synthetic image generation [16], ensembles based on architectures such as ResNet50 and MobileNetV2 [17], and transformer-based models such as DeiT [18]. These developments demonstrate the rapid evolution of image-based agricultural diagnosis, while also increasing the importance of establishing transparent and reproducible baselines against which more complex architectures can be evaluated.

Research conducted in Latin American and Spanish-speaking contexts has also explored artificial intelligence and computer vision for agricultural disease recognition.

Recent contributions have investigated deep learning for tomato leaf disease detection [19], artificial intelligence-based identification of tomato diseases [20], convolutional neural networks for foliar disease recognition [21], artificial vision and machine learning for early identification of tomato pests and diseases [22], SVM-based classification of diseased tomato leaves [23], and intelligent computer-vision systems for disease detection in hydroponic crops [24]. These studies demonstrate growing regional interest in intelligent agricultural technologies and reinforce the need for reproducible experimental frameworks that can subsequently be evaluated using locally acquired field data.

Despite the extensive literature on plant disease classification, direct comparisons between classical machine learning and transfer learning are frequently affected by differences in dataset partitions, preprocessing strategies, balancing procedures, feature representations, and evaluation criteria. Consequently, differences in reported accuracy cannot always be attributed exclusively to the classification algorithm. A controlled comparison of KNN, SVM, Random Forest, and MobileNetV2 using the same training, validation, and test partitions therefore provides useful evidence regarding the trade-off between predictive performance and methodological complexity.

The present study addresses this issue through a reproducible comparative framework for tomato leaf disease classification using the PlantVillage dataset. Classical models are represented by KNN, SVM, and Random Forest using a common handcrafted feature vector composed of HSV color histograms, gray-level co-occurrence matrix (GLCM) descriptors, and local binary patterns (LBP). These models are compared with a MobileNetV2 transfer learning architecture using the same dataset partition and evaluation protocol. Class balancing is performed exclusively on the training subset, preserving the original distributions of the validation and test subsets to reduce the risk of information leakage.

The main contribution of this study is therefore not the introduction of a new classification architecture, but the establishment of a controlled and reproducible comparison between feature-based classical machine learning and transfer learning under equivalent experimental conditions. Particular attention is given to predictive performance, class-level error patterns, reproducibility, and computational considerations. The results are interpreted as a methodological baseline for controlled imaging conditions rather than evidence of field-ready diagnostic performance, providing a foundation for subsequent validation using images acquired in real agricultural environments.

2 Materials and Methods

2.1 Research Design

This study adopted a quantitative, experimental, and comparative research design to evaluate classical machine learning and transfer learning approaches for tomato leaf disease classification. The experimental framework was designed to ensure that all evaluated models were compared under consistent data partitions, preprocessing conditions, model-selection criteria, and performance metrics.

Four classification strategies were investigated: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and a convolutional neural network based on MobileNetV2 transfer learning. The first three models used handcrafted image descriptors, whereas MobileNetV2 learned discriminative representations directly from image tensors. The comparison was therefore structured to assess whether a compact feature-based representation combined with classical classifiers could provide competitive performance relative to a pretrained deep learning architecture under controlled image conditions.

Particular attention was given to experimental reproducibility and prevention of information leakage. The original image dataset was partitioned before any balancing operation was performed. Image augmentation was subsequently applied only to the training subset, while validation and test subsets retained their original distributions. Hyperparameter selection and model configuration decisions were based exclusively on the training and validation subsets, and the test subset was reserved for final model evaluation.

2.2 Dataset

Experiments were conducted using the tomato subset of the PlantVillage dataset, a publicly available image repository widely used as a benchmark for automated plant disease recognition [1]. The original, non-augmented collection contained 18,178 tomato leaf images distributed across ten categories: bacterial spot, early blight, late blight, leaf mold, Septoria leaf spot, spider mites, target spot, tomato yellow leaf curl virus, tomato mosaic virus, and healthy leaves.

The dataset was partitioned before any augmentation was applied. Within each class, filenames were sorted, shuffled using a fixed random seed (`SEED`=42), and assigned approximately 70% to training and 15% to validation; the remaining observations were assigned to the test subset. This class-wise procedure produced 12,720 original training images, 2,722 validation images, and 2,736 test images. Validation and test subsets were kept unchanged throughout model development.

After partitioning, only the training subset was balanced by augmentation. Each training class was expanded to 3,749 images, corresponding to the largest class in the original training partition. The final balanced training set therefore contained 37,490 images, while the validation and test subsets retained their original class distributions. Table 1 distinguishes the original training counts from the balanced counts used for model fitting.

2.3 Training-Set Balancing and Image Augmentation

The original training partition was imbalanced, ranging from 261 tomato mosaic virus images to 3,749 tomato yellow leaf curl virus images. To reduce the influence of this imbalance during model fitting, oversampling by image augmentation was applied exclusively to the training subset after the original data split.

For each synthetic sample, one original image from the corresponding training class was selected at random and exactly one of five transformations was applied: horizontal flipping, vertical flipping, rotation within $\pm 20^{\circ}$, brightness scaling by a factor sampled

Table 1 Distribution of original and balanced tomato leaf images across the experimental subsets.

Class	Orig. train	Balanced train	Validation	Test	Orig. total
Bacterial spot	1488	3749	319	320	2127
Early blight	700	3749	150	150	1000
Late blight	1336	3749	286	287	1909
Leaf mold	668	3749	143	144	955
Septoria leaf spot	1239	3749	265	267	1771
Spider mites	1173	3749	251	252	1676
Target spot	982	3749	210	212	1404
Tomato yellow leaf curl virus	3749	3749	803	805	5357
Tomato mosaic virus	261	3749	55	57	373
Healthy	1124	3749	240	242	1606
Total	**12720**	**37490**	**2722**	**2736**	**18178**

between 0.7 and 1.3, or a central crop obtained by removing a 10% margin followed by resizing to the original spatial dimensions. Synthetic samples were generated until each training class contained 3,749 images.

No synthetic image was introduced into the validation or test subsets. Thus, augmentation modified only the distribution used for model fitting and did not alter the observations used for model selection or final performance estimation.

2.4 Image Preprocessing

All images were resized to 224×224 pixels. Model-specific preprocessing was then applied because the classical feature-based branch and the pretrained MobileNetV2 branch required different input representations.

For KNN, SVM, and Random Forest, images were read as RGB arrays and no global rescaling of raw pixel intensities to $[0, 1]$ was applied before handcrafted feature extraction. Instead, normalization was performed at the descriptor level: each HSV histogram and the LBP histogram were divided by their respective total frequencies. The four GLCM statistics were concatenated with these normalized histograms without additional pixel-level normalization. For SVM only, the resulting 110-dimensional feature vectors were standardized using `StandardScaler`, fitted on the training data and subsequently applied to validation and test features.

For MobileNetV2, images were loaded in mini-batches and transformed using the architecture-specific `preprocess_input` function from TensorFlow/Keras. This operation maps image intensities to the numerical range expected by the ImageNet-pretrained MobileNetV2 model. No additional online augmentation was applied by the image generator because the balanced training images had already been materialized during the preceding augmentation stage.

2.5 Handcrafted Feature Extraction

KNN, SVM, and Random Forest were trained using numerical descriptors extracted from each image rather than raw pixel arrays. A 110-dimensional feature vector was

Table 2 Handcrafted image descriptors used by the classical machine learning models.

Descriptor	Configuration and information represented	Dimension
HSV histograms	32 bins for each of the hue, saturation, and value channels; normalized color-distribution descriptors.	96
GLCM statistics	Contrast, correlation, energy, and homogeneity from grayscale images using distance 1 and angle 0°.	4
Uniform LBP	$P = 8$, $R = 1$, uniform mapping; normalized 10-bin local micro-texture histogram.	10
Total	–	**110**

constructed by combining color and texture descriptors derived from three complementary techniques: Hue–Saturation–Value (HSV) histograms, gray-level co-occurrence matrix (GLCM) statistics, and local binary patterns (LBP).

Color information was represented through independent histograms for the hue, saturation, and value channels. Each channel was discretized into 32 bins, producing 96 color features per image. These descriptors characterize chromatic properties that may be associated with foliar symptoms such as chlorosis, necrosis, discoloration, and heterogeneous pigmentation.

Texture information was characterized using GLCM descriptors computed from grayscale images. A pixel distance of one and an orientation of 0° were used. Four statistical properties were extracted: contrast, correlation, energy, and homogeneity.

Local micro-texture was represented using uniform LBP with $P = 8$ neighboring pixels and radius $R = 1$. The resulting uniform LBP histogram contributed ten additional features.

The final feature vector therefore contained 110 attributes:

$$32_{\text{Hue}} + 32_{\text{Saturation}} + 32_{\text{Value}} + 4_{\text{GLCM}} + 10_{\text{LBP}} = 110.$$

Table 2 summarizes the handcrafted descriptors.

HSV histograms and the LBP histogram were normalized by their respective total frequencies. The four GLCM statistics were then concatenated with the normalized HSV and LBP descriptors to form the final 110-dimensional representation used by KNN, SVM, and Random Forest.

2.6 Classification Models

Four supervised classification approaches were evaluated to represent different learning paradigms: instance-based learning through KNN, margin-based nonlinear classification through SVM, ensemble learning through Random Forest, and transfer learning through MobileNetV2.

2.6.1 K-Nearest Neighbors

KNN assigns the class of an unseen observation according to the predominant class among its k nearest samples in feature space. Euclidean distance was used as the similarity measure.

Four candidate neighborhood sizes were evaluated:

$$k \in \{3, 5, 7, 9\}.$$

The optimal value of k was selected according to the highest macro F1-score obtained on the validation subset. The selected configuration was subsequently evaluated once on the independent test subset.

2.6.2 Support Vector Machine

SVM was evaluated using the 110-dimensional handcrafted feature vectors. Because differences in feature scale can affect margin-based optimization, the features were standardized to zero mean and unit variance using `StandardScaler`. The scaler was fitted exclusively on the training features and then applied unchanged to validation and test features.

Two kernels were compared: linear and radial basis function (RBF). In both cases, the regularization parameter was fixed at $C = 10$; for the RBF model, gamma used the `scale` setting. The implementation used scikit-learn `SVC` with `decision_function_shape='ovr'`. Although this option returns an OVR-shaped decision function, `SVC` retains its underlying one-versus-one multiclass training formulation. Kernel selection was based on validation macro-F1.

2.6.3 Random Forest

Random Forest was implemented as an ensemble of 200 decision trees. Individual trees were constructed using random samples from the training observations and random subsets of the available features, and the final class prediction was obtained through majority voting.

Maximum tree depth was left unrestricted, allowing individual trees to grow according to the available training information. Feature standardization was not applied because tree-based classifiers are invariant to monotonic rescaling of the input variables.

2.6.4 MobileNetV2 Transfer Learning

The deep learning model was based on MobileNetV2 pretrained on ImageNet [25]. The convolutional base was connected to a task-specific head composed of global average

pooling, a fully connected layer with 256 rectified linear units, dropout with a rate of 0.4, and a ten-unit softmax output layer.

Training followed two consecutive stages. In the first stage, the complete MobileNetV2 base was frozen and called with `training=False`, while only the classification head was optimized. Adam with a learning rate of 10^{-3} was used for a maximum of ten epochs. Early stopping monitored validation accuracy with a patience of five epochs and restored the best weights. A separate `ReduceLROnPlateau` callback monitored validation loss with a patience of three epochs and a reduction factor of 0.5.

In the second stage, the convolutional base was made trainable and only its final 30 layers were considered for fine-tuning; all preceding layers remained frozen. BatchNormalization layers within the fine-tuned block were also kept frozen to preserve stable pretrained normalization statistics. The model was recompiled with Adam using a learning rate of 10^{-5} and trained for a maximum of ten additional epochs. New callback instances were created for this stage so that patience counters did not carry over from the frozen-base phase. The second-stage learning-rate scheduler used the same reduction factor and patience values.

The MobileNetV2 branch used architecture-specific ImageNet preprocessing through `preprocess_input`. Global seeds were reset before model construction and training to improve reproducibility.

2.7 Hyperparameter Selection Protocol

Hyperparameter selection was conducted exclusively using the validation subset. The independent test set was not used during model configuration, balancing, or model selection.

For KNN, the number of neighbors was selected from $k = \{3, 5, 7, 9\}$. For SVM, linear and RBF kernels were compared. Random Forest used a predefined configuration of 200 trees, whereas MobileNetV2 followed the two-stage transfer learning procedure described above.

The evaluated configurations are summarized in Table 3.

2.8 Experimental Workflow

The overall experimental procedure is illustrated in Figure 1. The workflow began with the selection of the tomato subset of PlantVillage, followed by image resizing and class-wise separation into training, validation, and test subsets. Training data were subsequently balanced through augmentation without modifying the validation and test distributions.

After the common preprocessing stage, the experiment was divided into two computational branches. In the classical machine learning branch, HSV, GLCM, and LBP descriptors were extracted and concatenated into a 110-dimensional vector used to train KNN, SVM, and Random Forest. In the transfer learning branch, resized image tensors were provided directly to MobileNetV2.

Model configurations were selected using validation performance. The final selected model from each family was then evaluated on the independent test subset using a

Table 3 Model configurations and hyperparameter-selection criteria.

Model	Parameter	Evaluated value(s)	Selection criterion
KNN	Number of neighbors	$\{3, 5, 7, 9\}$	Highest validation macro F1-score
KNN	Distance metric	Euclidean	Fixed
SVM	Kernel	Linear, RBF	Highest validation macro F1-score
SVM	C	10	Fixed
SVM	Gamma	Scale	Fixed
SVM	Decision-function shape	OVR	Fixed
Random Forest	Number of trees	200	Fixed
Random Forest	Maximum depth	None	Fixed
MobileNetV2	Training epochs	Maximum $10 + 10$	Validation monitoring
MobileNetV2	Initial learning rate	0.001	Fixed
MobileNetV2	Fine-tuning learning rate	1×10^{-5}	Fixed
MobileNetV2	Dropout	0.4	Fixed
MobileNetV2	Unfrozen layers	Final 30 layers	Fixed

common set of classification metrics. Performance was analyzed both globally and at the individual class level through normalized confusion matrices.

2.9 Performance Evaluation

All final models were evaluated using the same independent test subset containing 2736 images. Performance was quantified using accuracy, precision, recall, and F1-score. Because the classification problem contained classes with different numbers of original observations, macro-averaged metrics were emphasized so that each category contributed equally to the aggregate performance measurement. Weighted F1-score was additionally calculated to account for class support.

For a binary classification problem, accuracy can be expressed as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Precision is defined as

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2}$$

and recall as

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3}$$

The F1-score corresponds to the harmonic mean of precision and recall:

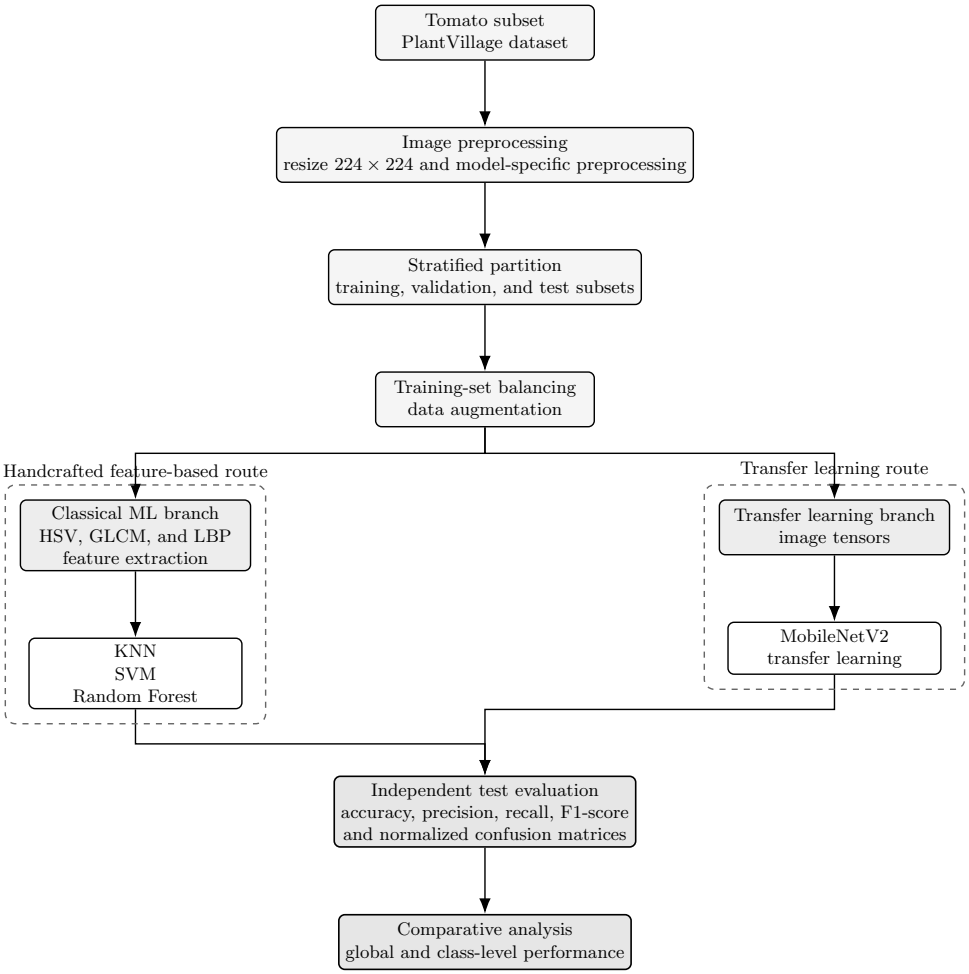


Fig. 1 Experimental workflow used for the comparative evaluation of classical machine learning and transfer learning models. Data augmentation was restricted to the training subset, whereas validation and test subsets retained their original distributions.

$$\mathrm{F1} = 2\frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{4}$$

For the multiclass classification problem considered in this study, precision, recall, and F1-score were first calculated independently for each disease class. Macro averages were subsequently computed as the unweighted mean across the ten classes.

In addition to aggregate metrics, row-normalized confusion matrices were generated for each classifier. This analysis enabled the identification of class-specific error patterns and visually similar disease categories that were more frequently confused by the models.

To quantify uncertainty in the final test estimates, 95% Wilson confidence intervals were calculated for accuracy. In addition, stratified nonparametric bootstrap confidence intervals were estimated using 2,000 resamples with a fixed random seed of 42. Resampling was performed independently within each true class so that the observed test-set class composition was preserved. Percentile 95% confidence intervals were obtained for accuracy, macro-precision, macro-recall, and macro-F1.

Because all four classifiers were evaluated on exactly the same test images, paired differences in classification correctness were assessed using exact two-sided McNemar tests. The six possible model pairs were tested, and Holm correction was applied to control the family-wise error rate at $\alpha = 0.05$. Before paired testing, file order and true-label order were programmatically verified to be identical between the classical and MobileNetV2 evaluation routes.

2.10 Reproducibility and Leakage Control

Several measures were adopted to improve experimental reproducibility and reduce the possibility of optimistic performance estimates. First, the training, validation, and test partitions were generated before data augmentation. Second, augmentation was restricted to the training subset. Third, model and hyperparameter selection relied exclusively on training and validation information. Fourth, the test subset remained isolated until the final evaluation stage.

A fixed random seed was used for dataset partitioning, and random seeds were also controlled in Python, NumPy, and TensorFlow where applicable. The same validation and test subsets were employed for all four classification strategies.

These procedures were particularly important because synthetic variants created through data augmentation may retain visual characteristics of their source images. Separating the original data before augmentation reduces the possibility that transformations derived from the same source image appear in both training and evaluation subsets.

2.11 Computational Environment

The experiments were conducted in Google Colaboratory using an NVIDIA T4 GPU accelerator. Classical machine learning models were implemented with scikit-learn version 1.2 or later. Image preprocessing was performed using OpenCV version 4.7 or later, while GLCM and LBP descriptors were obtained using scikit-image version 0.20 or later.

The MobileNetV2 architecture and transfer learning procedure were implemented using TensorFlow/Keras version 2.12 or later. Global random seeds were fixed in Python, NumPy, and TensorFlow to improve reproducibility across executions.

The final two-stage MobileNetV2 experiment required approximately 73.93 minutes of total training time in the reported computational environment. This value is reported as an implementation reference rather than as a hardware-independent benchmark, since execution time may vary according to GPU availability, software configuration, batch processing, and computational load.

The resulting experimental framework provides a consistent basis for comparing classical feature-based models and transfer learning under controlled imaging conditions while maintaining an explicit separation between model development, validation, and independent final testing.

3 Results

3.1 Validation-based model selection

Model selection was performed exclusively on the validation set. For KNN, the evaluated neighborhood sizes were $k = \{3, 5, 7, 9\}$. The best validation performance was obtained with $k = 3$, yielding an accuracy of 55.62% and a macro-F1 score of 52.40%. Performance decreased for larger neighborhood sizes, with macro-F1 scores of 50.19%, 49.68%, and 48.10% for $k = 5$, $k = 7$, and $k = 9$, respectively.

For SVM, the RBF kernel outperformed the linear kernel on the validation set. The linear configuration achieved 91.66% accuracy and 90.85% macro-F1, whereas the RBF configuration achieved 97.32% accuracy and 97.04% macro-F1. The RBF configuration was therefore retained for final evaluation. Random Forest achieved 95.70% validation accuracy and 95.26% macro-F1.

3.2 MobileNetV2 training behavior

MobileNetV2 was trained in two consecutive stages. During frozen-base training, validation accuracy increased from 91.00% in the first epoch to a maximum of 95.04% at epoch 7, whose weights were restored at the end of the first stage. During fine-tuning, validation accuracy reached 97.06% at epoch 10. The minimum validation loss during the fine-tuning stage was 0.1002 at epoch 6. At the final epoch, training accuracy was 99.78% and validation accuracy was 97.06%.

Figure 2 shows the training and validation trajectories across both stages.

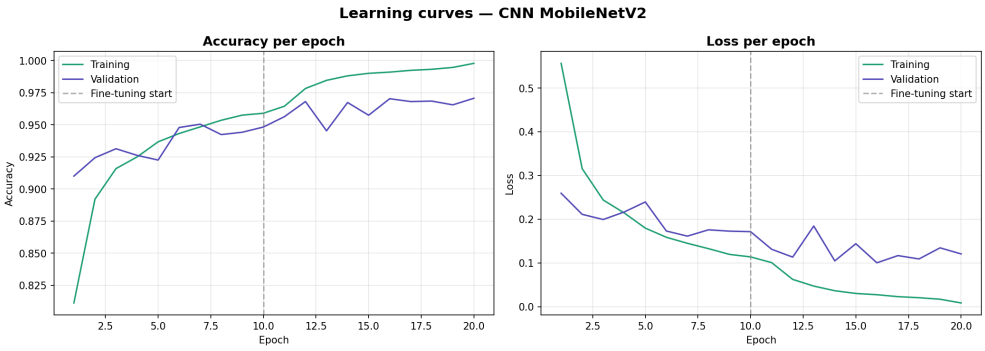


Fig. 2 Training and validation accuracy and loss for the two-stage MobileNetV2 transfer-learning procedure. The dashed vertical line indicates the transition from frozen-base training to fine-tuning.

Table 4 Final performance on the independent test set ($n = 2,736$).

Model	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	95% CI Accuracy (%)
KNN	58.22	54.09	57.06	54.35	56.37–60.06
SVM-RBF	97.11	96.48	96.47	**96.47**	96.42–97.68
Random Forest	96.93	96.17	**96.69**	96.42	96.21–97.51
MobileNetV2	**97.26**	**96.53**	96.04	96.24	96.58–97.81

3.3 Performance on the independent test set

Table 4 summarizes final performance on the held-out test set of 2,736 images. KNN achieved 58.22% accuracy and 54.35% macro-F1. In contrast, SVM-RBF, Random Forest, and MobileNetV2 all achieved accuracies close to 97%.

MobileNetV2 produced the highest numerical accuracy (97.26%) and macro-precision (96.53%). SVM-RBF achieved the highest macro-F1 (96.47%), whereas Random Forest obtained the highest macro-recall (96.69%). Thus, no single high-performing model ranked first across all aggregate metrics.

The 95% Wilson intervals for accuracy were 56.37–60.06% for KNN, 96.42–97.68% for SVM-RBF, 96.21–97.51% for Random Forest, and 96.58–97.81% for MobileNetV2. The intervals for the three high-performing models overlapped substantially (Figure 3).

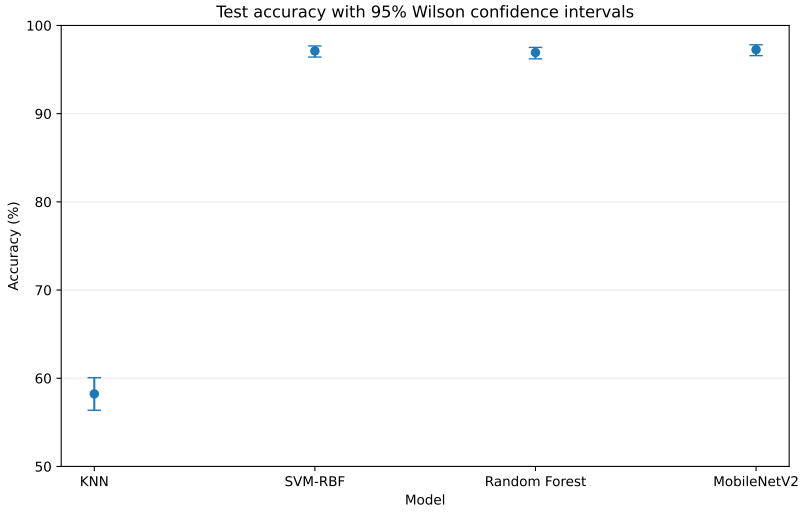


Fig. 3 Test-set accuracy with 95% Wilson confidence intervals for the four evaluated models.

Table 5 Exact McNemar pairwise comparisons with Holm correction.

Comparison	A correct/B wrong	A wrong/B correct	Exact p	Holm p
KNN vs. SVM-RBF	24	1088	< 0.001	< 0.001
KNN vs. Random Forest	25	1084	< 0.001	< 0.001
KNN vs. MobileNetV2	25	1093	< 0.001	< 0.001
SVM-RBF vs. Random Forest	49	44	0.679	1.000
SVM-RBF vs. MobileNetV2	65	69	0.796	1.000
Random Forest vs. MobileNetV2	65	74	0.498	1.000

Stratified bootstrap analysis with 2,000 resamples showed a similar pattern. The 95% bootstrap interval for macro-F1 was 52.36–56.40% for KNN, 95.66–97.29% for SVM-RBF, 95.60–97.16% for Random Forest, and 95.33–97.10% for MobileNetV2.

3.4 Paired statistical comparison

Exact McNemar tests were used to compare paired classification outcomes because all models were evaluated on the same test images. Holm correction was applied across the six pairwise comparisons. KNN differed significantly from each of the other models after multiplicity correction ($p_{\text{Holm}} < 0.001$). By contrast, none of the paired comparisons among SVM-RBF, Random Forest, and MobileNetV2 was statistically significant.

For SVM-RBF versus Random Forest, the exact p-value was 0.679; for SVM-RBF versus MobileNetV2, $p = 0.796$; and for Random Forest versus MobileNetV2, $p = 0.498$. The Holm-adjusted p-value was 1.000 for each of these three comparisons. Accordingly, the small differences in overall accuracy among these models were not supported as statistically significant differences in paired classification correctness.

3.5 Class-specific performance and confusion patterns

The class-wise results showed different performance profiles for the three strongest models rather than a uniform ranking across all categories. Figure 4 presents per-class F1 scores for SVM-RBF, Random Forest, and MobileNetV2.

MobileNetV2 achieved F1 scores of 98.75% for bacterial spot, 97.08% for late blight, and 99.50% for tomato yellow leaf curl virus. SVM-RBF achieved the highest F1 among the three models for early blight (91.95%), target spot (95.57%), and healthy leaves (99.79%). Random Forest achieved the highest F1 for leaf mold (98.27%), Septoria leaf spot (97.18%), spider mites (96.65%), and tomato mosaic virus (97.44%).

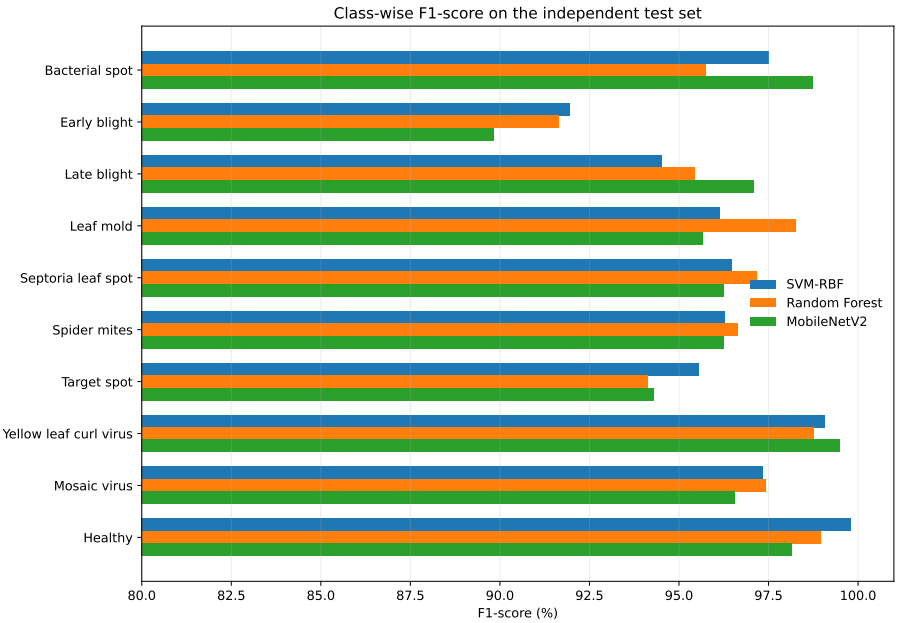


Fig. 4 Class-wise F1 scores for SVM-RBF, Random Forest, and MobileNetV2 on the independent test set.

The lowest MobileNetV2 recall occurred for early blight, with 85.33% (128 of 150 images) correctly classified. Among the 22 early-blight errors, 11 were assigned to late blight. MobileNetV2 recall was 99.06% for bacterial spot, 98.61% for late blight, 99.50% for tomato yellow leaf curl virus, 98.25% for tomato mosaic virus, and 99.59% for healthy leaves.

The normalized confusion matrices in Figure 5 show substantial off-diagonal dispersion for KNN and strongly diagonal structures for SVM-RBF, Random Forest, and MobileNetV2.

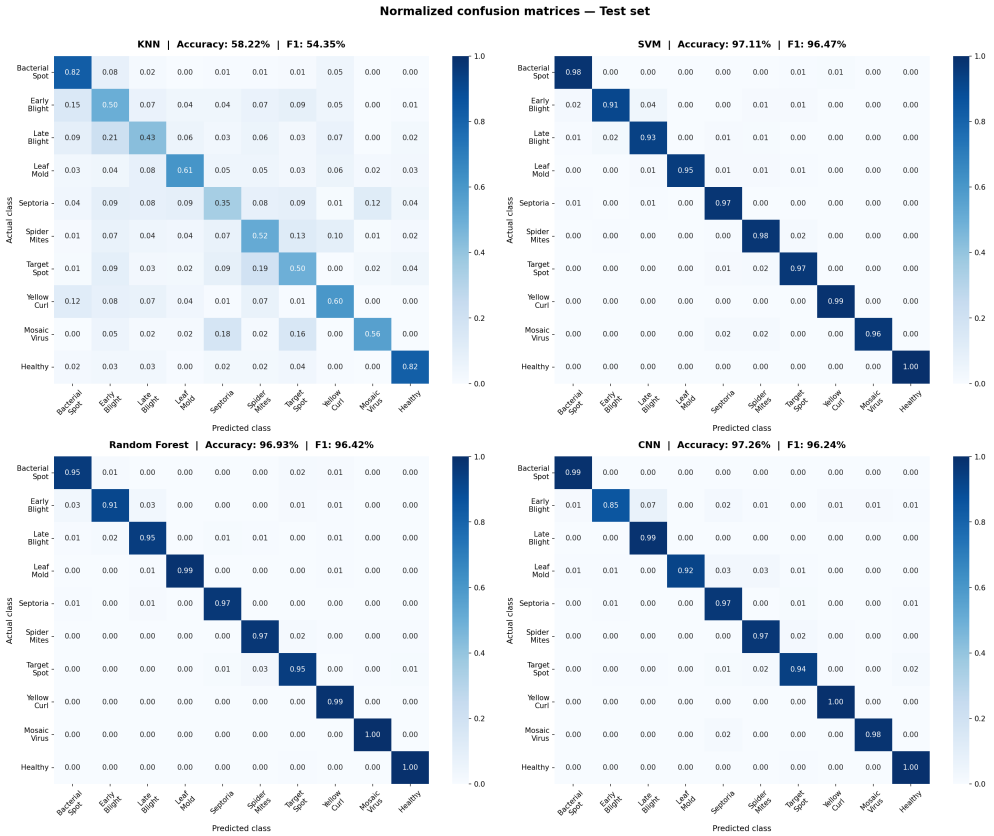


Fig. 5 Normalized confusion matrices for KNN, SVM-RBF, Random Forest, and MobileNetV2 on the independent test set. Values represent row-normalized proportions for each true class.

4 Discussion

The main result of this study was that three substantially different classification strategies—SVM-RBF, Random Forest, and MobileNetV2—reached closely comparable predictive performance under the controlled PlantVillage protocol. MobileNetV2 achieved the highest numerical accuracy, whereas SVM-RBF achieved the highest macro-F1 and Random Forest the highest macro-recall. More importantly, exact paired McNemar testing did not identify significant differences in classification correctness among these three models after Holm correction. The comparison therefore does not support a claim that the deep transfer-learning model was statistically superior to the two strongest classical alternatives on this test set.

The strong performance of SVM-RBF and Random Forest indicates that the 110-dimensional combination of HSV histograms, GLCM statistics, and uniform LBP retained sufficient color and texture information to discriminate most tomato leaf categories in this benchmark. This finding is consistent with the broader observation that

classical and deep approaches can both perform strongly when acquisition conditions are controlled and discriminative visual cues are relatively stable [11, 12]. The present study extends that comparison by holding the data partition, training-only balancing strategy, and independent test set constant across the model families.

MobileNetV2 nevertheless exhibited a competitive and distinct class-level error profile. Fine-tuning increased the best validation accuracy from 95.04% during frozen-base training to 97.06%, indicating that limited task-specific adaptation of the pretrained representation was beneficial. The model performed particularly well for bacterial spot, late blight, tomato yellow leaf curl virus, tomato mosaic virus, and healthy leaves. Its most evident weakness was early blight, where recall decreased to 85.33% and half of the errors were assigned to late blight. Because early and late blight may share lesion-level visual characteristics in individual images, this confusion pattern warrants further analysis with lesion localization or explainability methods rather than being interpreted solely from aggregate accuracy.

The class-specific results also demonstrate why a single global metric is insufficient for model selection. SVM-RBF, Random Forest, and MobileNetV2 ranked differently across disease categories, even though their overall accuracies were separated by less than 0.4 percentage points. In practical terms, the preferred model could therefore depend on the target disease distribution, acceptable error profile, deployment hardware, and whether interpretability or end-to-end feature learning is prioritized.

Computational comparisons require caution. The recorded prediction times for classical classifiers were measured after handcrafted features had already been extracted, whereas MobileNetV2 inference operated directly on image batches. Consequently, those timings are not equivalent end-to-end latency measurements and were not used to claim a deployment-speed advantage for any model. A fair efficiency comparison would need to include image loading, preprocessing, handcrafted feature extraction, and prediction under the same hardware and batch conditions.

Several limitations delimit the interpretation of these results. First, PlantVillage images are predominantly acquired under controlled conditions, with comparatively homogeneous backgrounds and limited field variability. High benchmark performance therefore cannot be interpreted as field-ready diagnostic accuracy. Second, the study used a single source dataset and did not include external validation from farms, mobile devices, or different geographic regions. Third, balancing relied on synthetic transformations of the original training images; although the split was performed before augmentation to prevent direct leakage, augmented samples remain correlated with their source images. Fourth, the MobileNetV2 experiment was evaluated under a deterministic single-seed training protocol; repeated independent training runs would provide additional information about optimization variability. Finally, the study addressed image-level classification and did not evaluate lesion localization, severity estimation, or multi-disease cases.

Future work should prioritize external field validation, repeated-seed experiments, and end-to-end computational benchmarking. Additional analyses using Grad-CAM or related explainability techniques could determine whether MobileNetV2 decisions are concentrated on biologically meaningful lesion regions. Domain adaptation and training with field-acquired images would also be necessary to assess robustness to

illumination changes, cluttered backgrounds, occlusion, variable leaf orientation, and camera heterogeneity.

5 Conclusion

This study established a controlled and reproducible comparison between three classical machine learning models and MobileNetV2 transfer learning for ten-class tomato leaf classification. On the independent PlantVillage test set, MobileNetV2 achieved the highest numerical accuracy (97.26%), SVM-RBF achieved the highest macro-F1 (96.47%), and Random Forest achieved the highest macro-recall (96.69%). Exact paired testing showed no statistically significant differences in classification correctness among these three models after Holm correction, while KNN performed substantially worse.

The results show that a compact handcrafted representation based on HSV, GLCM, and LBP can support classical nonlinear classifiers with predictive performance comparable to a pretrained deep network under controlled image conditions. This finding supports the use of classical baselines when evaluating more complex architectures and emphasizes that model selection should consider class-specific behavior and statistical uncertainty rather than accuracy alone. The conclusions remain limited to the controlled benchmark considered here; validation with independent field-acquired images is required before deployment-oriented claims can be made.

Acknowledgements. The authors gratefully acknowledge the Universidad de Guanajuato, Mexico, and the Corporación Universitaria Rafael Núñez, Colombia, for the partial institutional support provided during the development of this research. The authors also acknowledge the academic and technical support that facilitated the development, computational implementation, and evaluation of the proposed comparative framework.

Declarations

Funding. This research received partial institutional support from the Universidad de Guanajuato, Mexico, and the Corporación Universitaria Rafael Núñez, Colombia. No specific external grant from public, commercial, or not-for-profit funding agencies was awarded exclusively for this study.

Competing interests. The authors declare that they have no competing interests.

Ethics approval and consent to participate. Not applicable. The study used a publicly available image dataset and did not involve human participants or animals.

Consent for publication. Not applicable.

Data availability. The experiments were based on the publicly available PlantVillage dataset cited in this article. The exact train, validation, and test partitions, together with the derived prediction files used for the statistical analyses, are available from the corresponding author upon reasonable request.

Code availability. The source code used for image preprocessing, handcrafted feature extraction, model training, transfer learning, performance evaluation, and statistical analysis is available from the corresponding author upon reasonable request.

Author contributions. Conceptualization, J.M.B.S. and F.A.M.R.; methodology, J.M.B.S., F.A.M.R., and A.F.G.C.; software, F.A.M.R. and A.F.G.C.; validation, J.M.B.S., F.A.M.R., and A.F.G.C.; formal analysis, J.M.B.S., F.A.M.R., and A.F.G.C.; investigation, F.A.M.R. and A.F.G.C.; resources, J.M.B.S.; data curation, F.A.M.R. and A.F.G.C.; visualization, F.A.M.R. and A.F.G.C.; writing—original draft preparation, F.A.M.R. and A.F.G.C.; writing—review and editing, J.M.B.S., J.E.P.L., J.M.L.V., E.B.B., A.N.L.C., and A.L.M.H.; supervision, J.M.B.S. and A.L.M.H.; project administration, J.M.B.S. All authors reviewed and approved the final version of the manuscript.

References

[1] Hughes, D.P., Salathé, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. Preprint at https://arxiv.org/abs/1511.08060 (2016)

[2] Kumar, A., Vani, M.: Image based tomato leaf disease detection. In: Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1–6 (2019)

[3] Sharma, R., Panigrahi, A., Garanayak, M., Chakravarty, S., Paikaray, B., Pattanaik, L.: Tomato leaf disease detection using machine learning. In: Proceedings of ACI@ISIC, pp. 294–299 (2022)

[4] Jayanthi, G., Brindha, S., Vijayalakshmi, S., Dharshini, V., Freeda, J.A., Sahana, S.: Tomato leaf disease detection using machine learning. In: Proceedings of the 2024 International Conference on Communication, Computing and Internet of Things (IC3IoT), pp. 1–6 (2024)

[5] Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N.B., Koolagudi, S.G.: Tomato leaf disease detection using convolutional neural networks. In: Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, pp. 1–5 (2018)

[6] Ashok, S., Kishore, G., Rajesh, V., Suchitra, S., Sophia, S.G., Pavithra, B.: Tomato leaf disease detection using deep learning techniques. In: Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 979–983 (2020)

[7] Kolli, J., Vamsi, D.M., Manikandan, V.M.: Plant disease detection using convolutional neural network. In: Proceedings of the 2021 IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, pp. 1–6 (2021)

[8] Pandian, J.A., Kumar, V.D., Geman, O., Hnatiuc, M., Arif, M., Kanchanadevi, K.: Plant disease detection using deep convolutional neural network. Applied Sciences **12**, 6982 (2022)

[9] Shoaib, M., Hussain, T., Shah, B., Ullah, I., Shah, S.M., Ali, F., Park, S.H.: Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. Frontiers in Plant Science **13**, 1031748 (2022)

[10] Nawaz, M., Nazir, T., Javed, A., Masood, M., Rashid, J., Kim, J., Hussain, A.: A robust deep learning approach for tomato plant leaf disease localization and classification. Scientific Reports **12**, 18568 (2022)

[11] Tan, L., Lu, J., Jiang, H.: Tomato leaf diseases classification based on leaf images: A comparison between classical machine learning and deep learning methods. AgriEngineering **3**, 542–558 (2021)

[12] Pukkela, P., Borra, S.: Machine learning based plant leaf disease detection and severity assessment techniques: State-of-the-art. In: Classification in BioApps: Automation of Decision Making, pp. 199–226. Springer, Cham, Switzerland (2017)

[13] Al-gaashani, M.S., Shang, F., Muthanna, M.S., Khayyat, M., Abd El-Latif, A.A.: Tomato leaf disease classification by exploiting transfer learning and feature concatenation. IET Image Processing **16**, 913–925 (2022)

[14] Thangaraj, R., Anandamurugan, S., Kaliappan, V.K.: Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection **128**, 73–86 (2021)

[15] Pattnaik, G., Shrivastava, V.K., Parvathi, K.: Transfer learning-based framework for classification of pest in tomato plants. Applied Artificial Intelligence **34**, 981–993 (2020)

[16] Abbas, A., Jain, S., Gour, M., Vankudothu, S.: Tomato plant disease detection using transfer learning with c-gan synthetic images. Computers and Electronics in Agriculture **187**, 106279 (2021)

[17] Sharma, J., Al-Huqail, A.A., Almogren, A., Doshi, H., Jayaprakash, B., Bharathi, B., Hussen, S.: Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures. Scientific Reports **15**, 13904 (2025)

[18] Sun, C., Li, Y., Song, Z., Liu, Q., Si, H., Yang, Y., Cao, Q.: Research on tomato disease image recognition method based on DeiT. European Journal of Agronomy **162**, 127400 (2025)

[19] Arredondo-Basurto, E.R., Serrano-Talamantes, J.F., Olguín-Carbajal, M.,

Sandoval-Gutiérrez, J., Herrera-Lozada, J.C., Rivera-Zárate, I., Hernández-Bolaños, M.: Detección de enfermedades en hojas de tomate usando aprendizaje profundo. Computación y Sistemas **29** (2025)

[20] Moreno-Castillo, W., Acosta-Procel, J., Ruiz-Mesías, G., Vega-Iza, G., Muñoz-Espinoza, M.: Detección de enfermedades en hojas de tomate (*Solanum lycopersicum*) utilizando inteligencia artificial. Centro Agrícola **52**, 2462 (2025)

[21] Mora, E.A.H., Huitrón, V.G., Rangel, H.R., Sosa, L.E.A.: Detección de enfermedades foliares con arquitecturas de redes neuronales convolucionales. RINDERESU **5**, 1 (2021)

[22] Jácome García, J.F.: Detección temprana de minador, mosca blanca y fusarium en el tomate riñón, aplicando técnicas de visión artificial y machine learning. Bachelor Thesis, Universidad Nacional de Chimborazo, Riobamba, Ecuador (2022)

[23] Morante, J.S.P., Prieto, R.L.T.: Clasificación de hojas de tomate con plagas o enfermedades usando una máquina de soporte vectorial (SVM). Unpublished manuscript

[24] Morales, D.F.P., Dávalos, P.J.C.: Sistema inteligente para la detección de enfermedades en cultivos hidropónicos mediante visión artificial. CONECTIVIDAD **7**, 37–54 (2026)

[25] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryMaterialS1.zip