OPEN ACCESS
International Journal of Innovations in Science & Technology

Smart Farming with AI: Comparative Evaluation of CNN Models for Tomato Leaf Disease Classification

Syed Sohail Ahmed Shah[1], Mujeeb-u-Rehman Maree[2], Aftab Ahmed Chandio[2], Mujeeb-ur-Rehman Jamali[1]
[1]GC University Hyderabad, Sindh, Pakistan
[2]Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Sindh, Pakistan
*Correspondence: sohailahmed.shah@gcuh.edu.pk, mujeeb.jamali@gcuh.edu.pk, mujeeb@usindh.edu.pk, chandio.aftab@usindh.edu.pk

Citation | Shah. S. S. A, Maree. M. U. R Chandio. A. A Jamali. M. U. R, "Smart Farming with AI: Comparative Evaluation of CNN Models for Tomato Leaf Disease Classification", IJIST, Vol. 07, Issue. 04 pp 2787-2801, November 2025
Received | October 05, 2025 **Revised |** November 06, 2025 **Accepted |** November 13, 2025 **Published |** November 20, 2025.

Tomato is a major agricultural crop cultivated worldwide; however, its production is severely threatened by a wide range of plant diseases, necessitating accurate and timely detection methods. In recent years, neural network–based computer software and mobile applications have emerged as effective tools for plant disease detection. In this study, three advanced convolutional neural network (CNN) architectures—ResNet-50, DenseNet-121, and InceptionV3—are comparatively analyzed to evaluate their effectiveness in identifying and classifying tomato diseases using the PlantVillage dataset. To enhance model robustness against real-world variability, comprehensive image preprocessing and data augmentation techniques were employed, including rotation, horizontal and vertical flipping, rescaling, shear transformation, and zooming. A systematic hyperparameter tuning strategy was adopted by experimenting with various combinations of learning rates, batch sizes, and optimizers to optimize training performance. Experimental results demonstrate that hyperparameter optimization significantly improves classification accuracy, with the ResNet-50 model achieving the highest accuracy of 98.2%, along with superior F1-score, precision, and recall values. DenseNet-121 and InceptionV3 also exhibited strong performance, although their results were comparatively lower than those obtained with ResNet-50. These findings underscore the effectiveness of transfer learning and fine-tuning strategies in the development of automated systems for plant disease detection and classification. The study highlights the strong potential of CNN-based architectures for scalable and accurate disease detection, offering valuable support to farmers for early diagnosis and improved crop management. Furthermore, the study identifies future research directions, including deployment under real field conditions and the exploration of additional deep learning architectures.

Keywords: CNN, Deep Learning, Plant Disease Detection, Classification.

Introduction

Tomato is one of the most important agricultural crops worldwide, valued for its high nutritional content and significant economic contribution. It serves as a staple vegetable and a primary source of income for millions of farmers across the globe. Despite its importance, tomato production is severely constrained by a wide range of plant diseases, including bacterial spot, early blight, late blight, Septoria leaf spot, target spot, leaf mold, and viral infections such as Mosaic Virus and Yellow Leaf Curl Virus [1]. These diseases lead to substantial yield reductions and quality deterioration, thereby posing serious threats to food security and farmers' livelihoods.

Efficient crop management relies heavily on the accurate and timely identification of plant diseases to minimize losses. In many developing countries, disease diagnosis is still largely conducted through manual field inspections by agricultural experts using traditional methods. While widely practiced, these approaches are time-consuming, subjective, and impractical for large-scale or resource-constrained farming systems. Consequently, there has been growing interest in automating plant disease detection using deep learning and computer vision techniques, which enable accurate, scalable, and rapid analysis of plant leaf images [2].

Convolutional Neural Networks (CNNs) have emerged as one of the most effective technological advancements in plant disease identification and classification due to their exceptional ability to automatically extract discriminative features from image data. Well-established CNN architectures such as ResNet-50, DenseNet-121, and InceptionV3 have been extensively evaluated on benchmark datasets, including PlantVillage, consistently achieving classification accuracies exceeding 90%. Recent studies have further explored performance enhancements through hybrid CNN architectures, attention mechanisms, transfer learning with pre-trained weights, and image segmentation techniques combined with swarm intelligence methods [3].

Despite these advancements, several challenges persist. Model deployment in real-world field conditions and on edge devices remains limited due to the high computational and memory demands of deep CNN architectures [4]. Moreover, CNN performance is highly sensitive to hyperparameter selection, including learning rate, batch size, and optimizer choice, necessitating systematic tuning for optimal results. Additionally, models trained on controlled laboratory datasets such as PlantVillage often struggle to generalize to real field environments due to variations in background, illumination, and disease manifestation. The limited availability of standardized and comprehensive field datasets further exacerbates this issue, restricting the robust validation of model generalizability.

To address these challenges, this study conducts a comprehensive comparative analysis of three widely adopted CNN architectures—DenseNet-121, ResNet-50, and InceptionV3—for tomato leaf disease detection and classification. The study utilizes the publicly available PlantVillage dataset, which includes ten tomato leaf classes comprising nine disease categories and one healthy class. Extensive image preprocessing and data augmentation techniques—such as rotation, rescaling, width-wise and height-wise shifting, shearing, zooming, and horizontal flipping—are applied to enhance model robustness and generalization.

Furthermore, a systematic hyperparameter optimization strategy is implemented by exploring multiple combinations of optimizers, batch sizes, and learning rates to improve training efficiency and predictive accuracy. The models are trained and validated using stratified dataset splits, and their performance is evaluated using standard metrics including accuracy, precision, recall, and F1-score to ensure a comprehensive assessment of classification quality.

The main contributions of this study are as follows:

- A detailed comparative evaluation of DenseNet-121, ResNet-50, and InceptionV3 architectures for tomato leaf disease classification.

- Implementation of diverse data augmentation techniques to simulate realistic environmental and leaf appearance variations.
- Comprehensive hyperparameter tuning across learning rates, batch sizes, and optimizers to identify optimal training configurations.
- Demonstration of ResNet-50 achieving the highest overall classification accuracy (98.2%), highlighting its suitability for automated plant disease diagnosis.

Literature Review:

Recent studies have significantly advanced the application of deep learning techniques for plant disease detection, with a strong focus on improving CNN classification accuracy, refining training strategies, and addressing real-world deployment challenges. While CNN-based and hybrid deep learning models consistently report high classification performance, they also reveal limitations related to computational complexity, dataset constraints, and training instability.

A prominent research direction involves the development of advanced CNN architectures to enhance disease classification performance. For example, Vineel Pratap et al. [5] proposed an Integrated Hybrid ABOCNN model incorporating attention mechanisms and domain-specific knowledge, achieving 99.6% accuracy on rice leaf images. Comparable advancements have been reported in tomato disease studies, where DenseNet, EfficientNet, and modified AlexNet variants attained classification accuracies between 98% and 99% on large-scale datasets [6][7]. Despite their strong performance, these models often suffer from high memory consumption, increased inference time, and susceptibility to overfitting—limitations also observed in complex architectures such as CMCNN [8] and InceptionV3-based models [9].

Several studies have explored optimization strategies to improve CNN training efficiency and stability. For instance, DenseNet-121 combined with the AdaBound optimizer achieved an accuracy of 99.31% in plant disease classification [6], while PlantVillage-based research in [10] employed data augmentation, dropout, and regularization techniques to enhance model stability. However, these approaches remain sensitive to optimizer selection, learning rate adjustments, and convergence behavior, as evidenced by the unstable performance of ResNet-50 reported in [11].

Another stream of research integrates preprocessing and image segmentation techniques with deep learning models to improve feature extraction. Shrivastav et al. [12] combined Grasshopper Optimization with K-means clustering and CNN classification, achieving an accuracy of 97.6%. Similarly, the study in [13] employed grayscale conversion and contour-based enhancement, reporting superior performance of DenseNet-169 and Xception models compared to MobileNetV2. While such preprocessing pipelines can improve classification accuracy, they often introduce additional computational overhead and sensitivity to color and pixel-level variations.

Despite the high accuracies reported on laboratory-based datasets, significant practical challenges remain. Numerous studies emphasize the scarcity of field-acquired datasets, which limits the ability of models to generalize under real agricultural conditions [9]. Additionally, the computational demands of deep architectures restrict their deployment on resource-constrained devices such as smartphones and low-cost farm equipment. These challenges are consistently highlighted across high-performing CNN models [9][5][8], underscoring the gap between research performance and practical applicability.

Overall, the literature demonstrates that while deep learning has markedly improved plant disease detection, persistent challenges related to computational efficiency, training stability, and real-world generalization remain unresolved. Addressing these limitations requires approaches that balance high classification accuracy with reduced complexity and robust performance under diverse agricultural conditions.

Methodology:

This study employs the PlantVillage dataset, which comprises ten classes of tomato leaf images representing healthy and diseased samples. Image preprocessing includes resizing and normalization, with input dimensions set to 224 × 224 pixels for DenseNet-121 and ResNet-50, and 299 × 299 pixels for InceptionV3 to meet architectural requirements. Data augmentation techniques are applied to mitigate dataset imbalance and simulate real-world variability.

Three deep CNN architectures—DenseNet-121, ResNet-50, and InceptionV3—are trained and evaluated using multiple hyperparameter configurations. These configurations include varying learning rates, batch sizes, and optimizers (SGD, Adam, and RMSprop). Model implementation is conducted on Google Colab using a Tesla T4 GPU with 16 GB of GPU memory and 12 GB of system RAM. Performance evaluation is based on accuracy, precision, recall, and F1-score to identify the optimal architecture and hyperparameter combination.

The dataset is split into training, validation, and testing subsets using a 70:15:15 ratio. Stratified sampling ensures proportional representation of all disease classes across subsets. The training set is used for model learning, the validation set for hyperparameter tuning and model selection, and the test set for final performance evaluation. This fixed validation strategy ensures a consistent and fair comparison across models.

Dataset:

The PlantVillage dataset is a publicly available, large-scale collection of high-quality plant leaf images designed for automated plant disease diagnosis. It contains over 54,000 images spanning 14 crop species and provides a standardized benchmark for training and evaluating machine learning models. For tomato crops, the dataset includes ten classes: one healthy class and nine disease classes—Mosaic Virus, Early Blight, Late Blight, Leaf Mold, Yellow Leaf Curl Virus, Septoria Leaf Spot, Target Spot, Spider Mite, and Tomato Bacterial Spot.

Each image typically contains a single leaf captured under controlled conditions, which minimizes background clutter and enhances classification accuracy. While this controlled environment aids model training, it also limits direct generalization to field conditions. Nevertheless, the PlantVillage dataset remains a critical resource for benchmarking CNN-based plant disease detection models and serves as an essential foundation for advancing agri-tech applications.

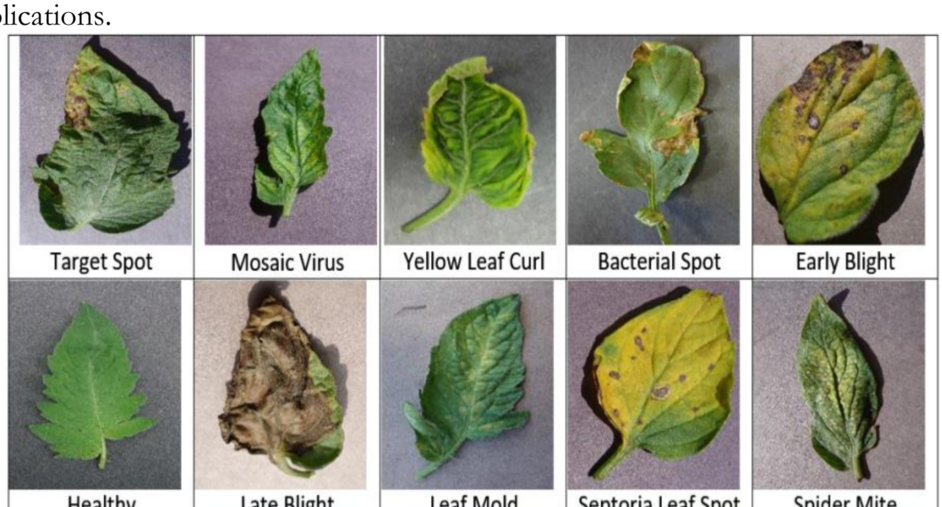

Figure 1. One healthy class and nine disease classes of Tomato plant leaf images of the Plant Village dataset.

The total number of images in the tomato category is 16050, having the following number of images in each disease class.

Table 1. Distribution of Tomato Plant Images

Class Name	Number of Images
Healthy	1604
Early blight	1000
Late blight	1910
Leaf Mold	952
Septoria Leaf Spot	1778
Spider mites	1676
Target Spot	1422
Yellow Leaf Curl Virus	3208
Mosaic Virus	373
Bacterial Spot	2127
Total Images	16050

Image Resizing and Normalization

The use of pre-trained models is a cornerstone of transfer learning, as models trained on large-scale real-world datasets such as ImageNet have demonstrated strong generalization capabilities across diverse computer vision tasks, including object detection and image classification. In this study, both DenseNet-121 and ResNet-50 architectures were employed using an input image size of **224 × 224 pixels**, which corresponds to the standard ImageNet configuration. This image resolution provides an optimal balance between feature representation, computational efficiency, and training performance, making it suitable for convolutional neural network (CNN)-based learning.

In contrast, the InceptionV3 architecture incorporates factorized convolutions and auxiliary classifiers, resulting in increased architectural complexity. Consequently, it requires a larger input image size of **299 × 299 pixels**. The higher spatial resolution enhances the model's ability to capture fine-grained features, which is particularly beneficial for complex classification tasks such as plant disease detection, where subtle visual differences between disease categories must be accurately identified. Therefore, for optimal performance, input image dimensions were carefully adapted to match the architectural requirements of each pre-trained model.

Image Augmentation:

To improve model robustness and generalization, comprehensive image augmentation techniques were applied during training. Image augmentation plays a crucial role in mitigating overfitting and enhancing a model's ability to perform effectively under real-world conditions. In this study, augmentation strategies included image rotation at varying angles, horizontal flipping, rescaling, width and height shifting, shear transformations, and zoom operations within predefined ranges.

These techniques artificially increased the diversity of the training dataset, enabling the CNN models to learn invariant features and improving their resilience to variations in orientation, scale, and illumination. As a result, the augmented dataset significantly enhanced the model's capacity to accurately identify and classify tomato plant diseases under diverse and challenging environmental conditions.

DenseNet-121 Architecture:

DenseNet-121 was effectively utilized for the detection and classification of tomato plant diseases due to its distinctive dense connectivity pattern. In this architecture, each layer receives feature maps from all preceding layers and passes its own feature maps to all subsequent layers. This design enhances gradient flow, promotes feature reuse, and mitigates the vanishing gradient problem, making DenseNet-121 particularly well suited for deep learning-based image classification tasks.

For tomato leaf disease detection, DenseNet-121 automatically extracts hierarchical and discriminative features that enable accurate differentiation between healthy leaves and multiple disease categories. The dense skip connections facilitate faster convergence and improved learning efficiency compared to traditional CNN architectures, leading to superior diagnostic performance.

Structurally, DenseNet-121 begins with an initial convolutional and pooling layer, followed by **four dense blocks** separated by transition layers. These transition layers reduce feature map dimensionality through convolution and pooling operations, while the dense blocks contain multiple convolutional layers with direct interconnections that encourage extensive feature reuse. The final feature representations are globally pooled and passed to a fully connected layer for multi-class disease classification.

Several studies have reported strong performance of DenseNet-121 in tomato plant disease detection, demonstrating high accuracy, precision, recall, and F1-score values. For instance, one study [12] trained DenseNet-121 on tomato leaf images transformed into the HSV color space and achieved classification accuracies ranging from **94% to 98%**, supported by effective preprocessing, thresholding, and data augmentation techniques. Furthermore, recent research highlights DenseNet-121's robustness and generalization capabilities, including its effectiveness in detecting rare or newly emerging diseases through the integration of novelty detection mechanisms [14][15]. These findings underscore the architecture's suitability for real-world deployment in automated plant disease detection systems, including mobile and real-time diagnostic platforms.

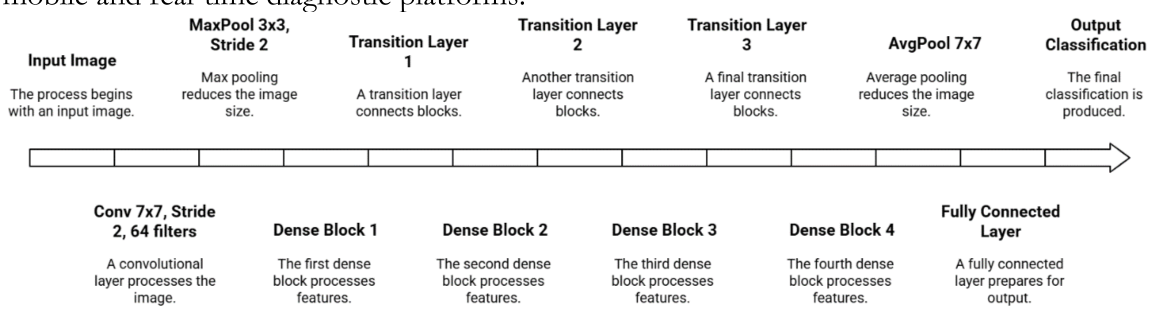

Figure 2. DenseNet-121 Architecture

ResNet-50

ResNet-50 is a deep convolutional neural network widely employed for tomato plant disease detection and classification due to its ability to efficiently train very deep architectures using **residual learning**. The model comprises 50 layers organized into a sequence of residual blocks, where **identity (skip) connections** bypass one or more layers. These connections mitigate the vanishing gradient problem, facilitate improved gradient flow, and enhance feature propagation throughout the network, enabling effective training of deep architectures.

In the context of tomato leaf disease detection, ResNet-50 automatically extracts complex hierarchical features, allowing the model to accurately differentiate between healthy leaves and various disease categories. Typically, a pre-trained ResNet-50 is fine-tuned on tomato leaf datasets, with extensive data augmentation applied to enhance generalization across variations in lighting, orientation, and leaf appearance.

Enhanced variants of ResNet-50 have been developed to improve feature extraction and classification performance. For example, replacing the standard 7×7 convolutional layer at the input with **cascaded atrous (dilated) convolutions** allows the model to capture leaf features at multiple scales. Integration of a **dual-path attention (DPA) mechanism** within residual blocks emphasizes key spatial and channel-wise features, improving disease identification by focusing on relevant leaf regions while suppressing background noise. In this variant, termed **ResNet50-DPA**, stochastic pooling is employed to reduce information loss compared to traditional pooling methods, and fully connected layers are replaced with **one-dimensional convolutions** to preserve more leaf-specific information. Visualization techniques such as **Grad-CAM** have demonstrated that ResNet50-DPA effectively highlights critical diseased regions on leaves, enhancing interpretability and diagnostic reliability [16].

From a performance perspective, ResNet-50 and its variants consistently achieve classification accuracies of **93–95%**, effectively recognizing multiple tomato leaf diseases regardless of disease severity or environmental conditions. The combination of **transfer learning** and fine-tuning enables ResNet-50 to perform robustly even with limited labeled datasets. These characteristics make ResNet-50 and its enhanced variants highly suitable for automated tomato disease detection, supporting early diagnosis and informed crop management strategies for farmers [17].

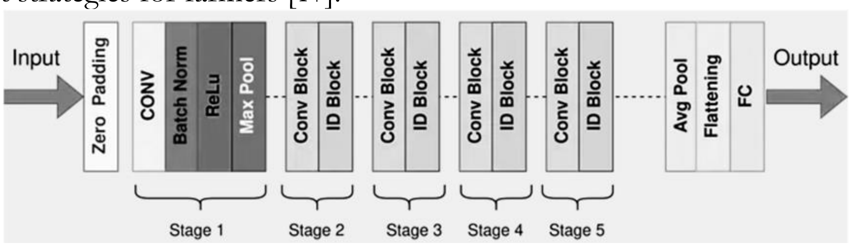

Figure 3. ResNet-50 Architecture.

InceptionV3:

The **InceptionV3** convolutional neural network is recognized for its computational efficiency and strong feature extraction capabilities, largely due to its **inception modules**. These modules perform convolutions of multiple kernel sizes in parallel, enabling the model to capture both fine and coarse features simultaneously. This multi-scale processing is particularly advantageous for detecting tomato leaf disease spots, which vary in size, shape, and appearance. InceptionV3 is thus highly effective in extracting rich hierarchical features from tomato leaf images, facilitating accurate classification and differentiation between healthy leaves and multiple disease categories.

The architecture comprises multiple **inception blocks**, each containing parallel convolutional layers with kernel sizes of 1×1, 3×3, and 5×5, along with pooling operations. This parallel structure allows the network to capture diverse spatial features efficiently. Additionally, InceptionV3 employs **factorized convolutions** for example, a 5×5 convolution is factorized into two 3×3 convolutions reducing computational cost without sacrificing feature representation. These architectural innovations improve training efficiency while maintaining high classification accuracy.

InceptionV3 can also be combined with other architectures or used independently for tomato leaf disease classification. Modifications, such as integrating Inception blocks into reduced 16-layer VGG networks, have demonstrated superior performance on the **PlantVillage tomato leaf dataset**, achieving accuracies exceeding **99%**. Such modifications enable the model to process multiple classes of diseases simultaneously, including bacterial spot, early blight, late blight, and healthy leaves, while maintaining computational efficiency [18].

Overall, InceptionV3's ability to capture multi-scale features in a resource-efficient manner makes it highly suitable for automated tomato disease detection, offering both high accuracy and practical feasibility for real-world deployment in agricultural settings.

Hyperparameter Optimization:

For tomato leaf disease detection and classification, careful **hyperparameter tuning** was performed to enhance model performance and generalization. Key hyperparameters considered in this study included the **learning rate**, **batch size**, and **optimizer**, with the number of epochs (Kappa) set to 3.

Three learning rates **0.01, 0.001, and 0.0001** were evaluated to examine their effect on model convergence and accuracy. To promote stable weight updates and improve generalization, three batch sizes **16, 32, and 64** were tested. Additionally, three widely used optimization algorithms **SGD, Adam, and RMSprop** were employed, each with learning rate and momentum parameters tailored to facilitate effective parameter updates during training.

In total, **nine experimental setups** were systematically implemented for each CNN model by combining these hyperparameters, allowing an assessment of the interactions among learning rate, batch size, and optimizer choice. This approach provided a structured framework for identifying the optimal hyperparameter configuration that maximized classification accuracy while maintaining robust generalization.

Overall, this hyperparameter optimization strategy offered a comprehensive and efficient methodology to evaluate model sensitivity to critical training parameters, ultimately improving the predictive performance and reliability of CNN-based tomato leaf disease detection.

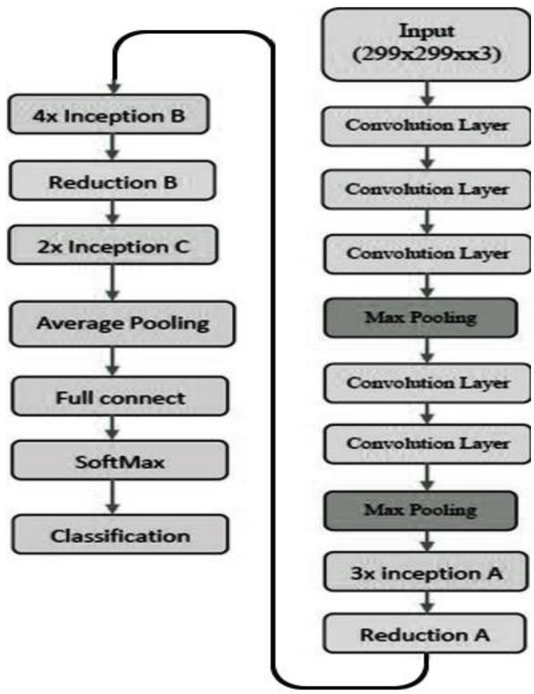

Figure 4. InceptionV3 Architecture

Augmentation Techniques:

Following the augmentation type and with given parameters, are used to get the optimized accuracy of the model.

Table 2. Parameters of Augmentation

Augmentation Type	Parameter	Effect / Purpose
Rescaling	rescale=1./255	It normalizes the pixel values from 0–255 to 0–1 for faster and more stable training.
Rotation	Range of Rotation=20	Randomly rotates the images by up to ±20°, which will help the model to recognize the diseases regardless of the leaf orientation.
Width Shift	Range of width shift=0.2	Shift the image horizontally by up to 20 percent of its width to show positional variation.

Height Shift	Range of height shift=0.2	Shift the image vertically by up to 20 percent of its height to enhance spatial robustness.
Shear Transformation	Range of shear=0.2	For random shearing, this gives perspective distortions training help for the CNN model.
Zoom	Range of zoom=0.2	Randomly zooming the images in or out by 20 percent, in this way, the model recognizes the image at different scales.
Horizontal Flip	Flip horizontal=True	Flips images horizontally. It is useful when leaf orientation does not affect the disease pattern.

Evaluation Metrics:

In the evaluation of a CNN model for tomato plant disease detection and optimization, several key metrics are commonly used to assess the model's classification performance comprehensively:

Accuracy:

This metric is used to measure the general performance of the model by evaluating the ratio of the correct classification of the samples (both diseased and healthy leaves) to the total number of samples. Although accuracy provides a general measure of the effectiveness of a model, it can be less accurate in case the dataset is skewed.

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

Precision:

The precision is used to measure the effectiveness of the model in identifying the positive cases (e.g., diseased leaves) correctly by determining the proportion of true positives to the number of positives predicted. The high precision implies that the model has low false identification cases of healthy leaves as diseased, which makes the correct estimation of the health of plants.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity):

In order to calculate recall, the model will perform on all the positive cases to compute the recall by determining the ratio of the number of positives correctly predicted to the total number of the total positives. The high recall signifies that the model is optimized such that it is more likely to detect many diseased leaves with minimal chances of false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score:

F1 score is the sum of the precision and the recall, and as a result, it is a combination of the two scores and, therefore, it represents a single score that strikes a balance between false positives and false negatives. It is mostly useful in cases where the accuracy and the recall are significant, like detecting a disease where the loss of a diseased leaf or false-alarming results might be influential.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Sample Standard Deviation:

The sample standard deviation is calculated from the standard formula given below, to check whether the best performing model, with an optimal combination of hyperparameters, is stable or not.

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

Where:

s = sample standard deviation

x_i = all data points
x bar = sample mean
n= total no. of data points in the sample

Experimental Results

This section summarizes the experimental results, highlighting the performance of deep convolutional neural network (CNN) models for tomato leaf disease detection and classification. The experiments are structured to compare the performance of three CNN architectures: DenseNet-121, ResNet-50, and InceptionV3. Extensive experiments were conducted, including **hyperparameter optimization**, application of multiple evaluation metrics, and detailed performance assessment, demonstrating the efficiency of the proposed models in accurately diagnosing and classifying diverse tomato leaf diseases. These results validate both the **robustness** and **generalization capability** of the developed approach, while also illustrating how different training configurations impact detection accuracy and overall system performance.

The study evaluated DenseNet-121, ResNet-50, and InceptionV3 under multiple hyperparameter combinations, including **learning rate, batch size, and optimizer**. Optimal configurations substantially improved model performance. The best-performing setups typically featured **learning rates of 1e-4 or 1e-3**, **batch sizes of 16 or 32**, and the **Adam optimizer**.

DenseNet-121 achieved a highest accuracy of **96.1%** using a learning rate of 1e-4, batch size of 32, and Adam optimizer, with precision, recall, and F1-score values of 95.7%, 95.3%, and 95.5%, respectively.

ResNet-50 reached the highest overall accuracy of **98.2%**, with precision, recall, and F1-score values of 97.6%, 97.3%, and 97.4%, respectively, under the same learning rate and batch size with Adam optimizer.

InceptionV3 achieved a maximum accuracy of **94.7%** with a learning rate of 1e-4, batch size of 32, and RMSprop optimizer, with precision, recall, and F1-score slightly below 94%.

All models exhibited decreased performance when trained with higher learning rates (1e-2) or using the SGD optimizer. Similarly, larger batch sizes (64) slightly reduced accuracy compared to smaller batches (16 or 32).

Experimental Setup and Evaluation:

All CNN models were implemented in **Python** using **TensorFlow** and **PyTorch** frameworks. Pre-trained models—DenseNet-121, ResNet-50, and InceptionV3—were loaded from official model libraries, while **NumPy, Pandas, and OpenCV** were used for data manipulation and preprocessing. All computations were performed on **Google Colab**, running on **Ubuntu Linux** with GPU acceleration.

Multiple evaluation metrics—including **accuracy, precision, recall, and F1-score**—were used to assess model performance comprehensively. During hyperparameter optimization, these metrics guided the selection of configurations that not only maximized overall accuracy but also balanced precision and recall, ensuring practical applicability in real agricultural settings.

To evaluate model **stability and variability**, each CNN was trained **three times** under its best hyperparameter configuration. The **sample standard deviation** of the resulting accuracies was calculated to quantify consistency and reliability across runs. Lower standard deviation values indicate stable performance despite factors such as random initialization and data shuffling. This analysis complements mean accuracy in assessing the robustness of CNN predictive capability.

Performance of Models

The performance of each CNN model was evaluated in two stages. First, **baseline models** were assessed to establish reference performance. Subsequently, each model was tested under **different hyperparameter combinations** to determine the optimal configuration for tomato leaf disease classification. The results provide insights into which CNN architecture is best suited for specific parameter settings, balancing accuracy, robustness, and computational efficiency.

Baseline Performance of DenseNet-121, ResNet-50, and InceptionV3 CNN Models without Optimization:

Table 3. Baseline Performance of DenseNet-121, ResNet-50, and InceptionV3

Model	Accuracy	Precision	Recall	F1-Score
DenseNet-121	89.5%	87.5%	86.5%	87.5%
ResNet-50	87.5%	85.5%	84.5%	85.5%
InceptionV3	86.5%	84.5%	83.5%	84.5%

Optimized DenseNet-121 Performance:

Table 4. Optimized DenseNet-121 Performance.

Combination	Learning Rate	Batch Size	Optimizer	Accuracy	Precision	Recall	F1-Score
1	1e-3	32	Adam	95.2%	94.8%	94.5%	94.6%
2	**1e-4**	**32**	**Adam**	**96.1%**	**95.7%**	**95.3%**	**95.5%**
3	1e-2	32	SGD	92.0%	91.0%	90.5%	90.7%
4	1e-3	16	Adam	95.8%	95.3%	95.0%	95.1%
5	1e-3	64	RMSprop	94.0%	93.5%	93.0%	93.2%
6	1e-4	16	SGD	93.5%	92.8%	92.3%	92.5%
7	1e-2	64	SGD	90.5%	89.5%	89.0%	89.2%
8	1e-3	64	Adam	95.5%	95.0%	94.7%	94.8%
9	1e-4	32	RMSprop	94.3%	93.8%	93.4%	93.6%

Optimized ResNet-50 Performance:

Table 5. Optimized ResNet-50 Performance.

Combination	Learning Rate	Batch Size	Optimizer	Accuracy	Precision	Recall	F1-Score
1	1e-3	32	Adam	94.5%	93.8%	93.5%	93.6%
2	**1e-4**	**32**	**Adam**	**98.2%**	**97.6%**	**97.3%**	**97.4%**
3	1e-2	32	SGD	90.8%	89.5%	89.0%	89.2%
4	1e-3	16	Adam	94.8%	94.2%	93.9%	94.0%
5	1e-3	64	RMSprop	93.5%	92.7%	92.3%	92.5%
6	1e-4	16	SGD	92.6%	91.8%	91.3%	91.5%
7	1e-2	64	SGD	89.0%	87.5%	87.0%	87.2%
8	1e-3	64	Adam	94.3%	93.6%	93.3%	93.4%
9	1e-4	32	RMSprop	93.8%	93.0%	92.6%	92.8%

Optimized InceptionV3 Performance:

Table 6. Optimized InceptionV3 Performance.

Combination	Learning Rate	Batch Size	Optimizer	Accuracy	Precision	Recall	F1-Score
1	1e-3	32	Adam	93.4%	92.8%	92.4%	92.6%
2	1e-4	32	Adam	94.3%	93.7%	93.4%	93.5%
3	1e-2	32	SGD	90.3%	89.3%	88.8%	89.0%
4	1e-3	16	Adam	93.9%	93.3%	92.9%	93.1%
5	1e-3	64	RMSprop	92.5%	91.8%	91.4%	91.6%

6	1e-4	16	SGD	91.8%	91.0%	90.5%	90.7%
7	1e-2	64	SGD	89.0%	87.9%	87.3%	87.6%
8	1e-3	64	Adam	93.6%	93.0%	92.6%	92.8%
9	**1e-4**	**32**	**RMSprop**	**94.7%**	**94.0%**	**93.6%**	**93.9%**

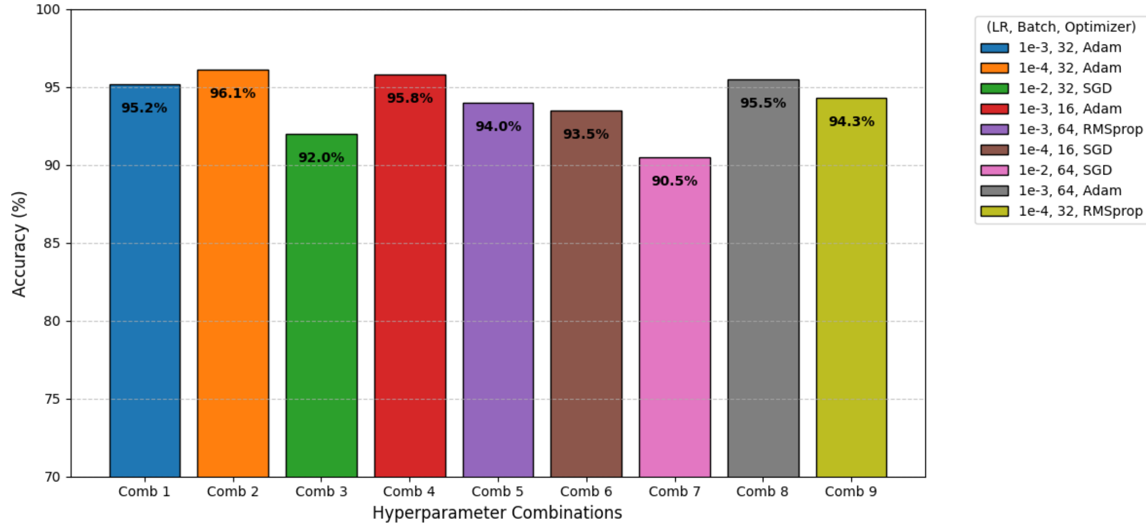

Figure 5. DenseNet-121 Hyperparameters Optimization Comparison.

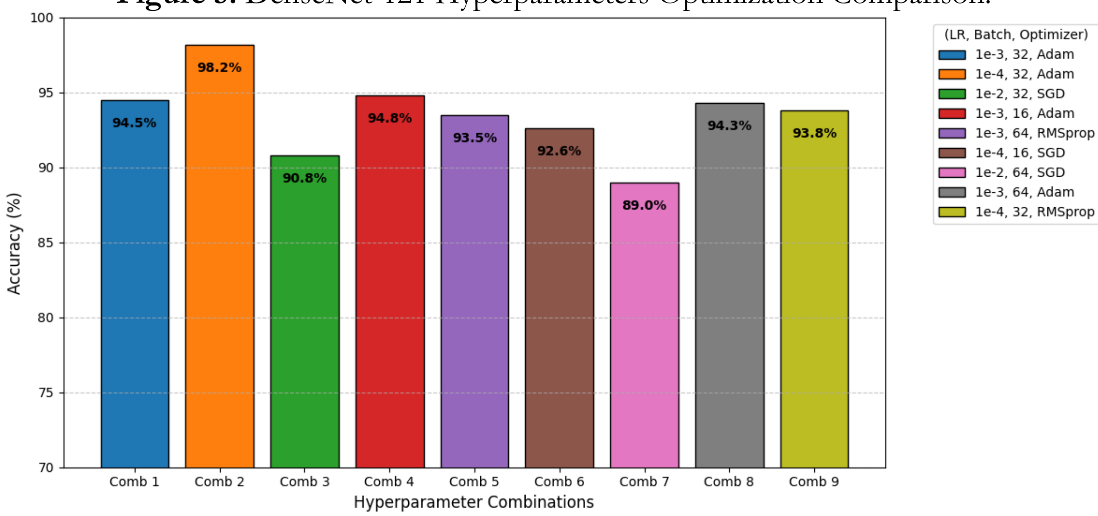

Figure 6. ResNet-50 Hyperparameters Optimization Comparison.

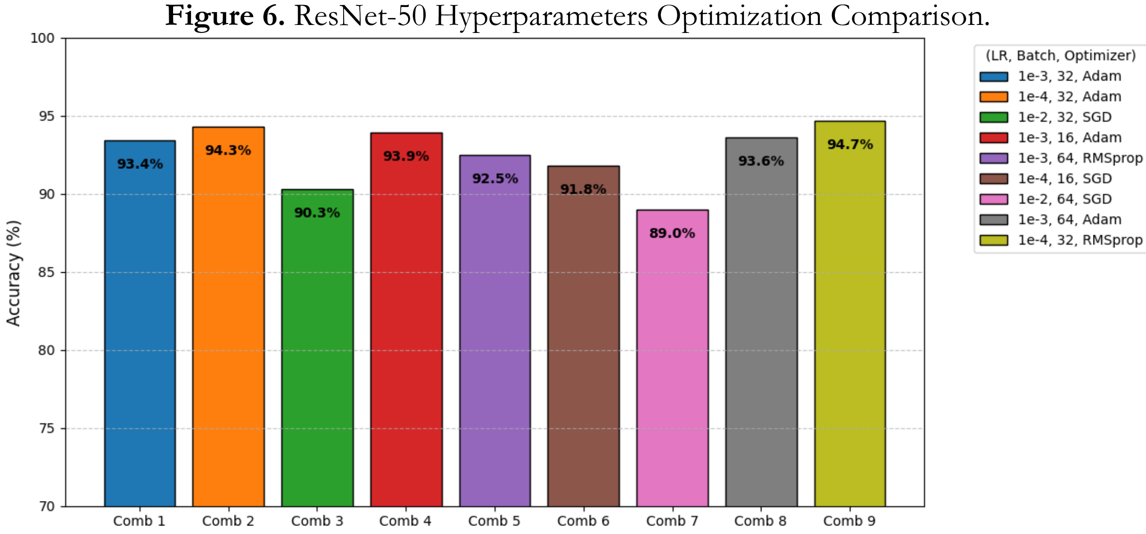

Figure 7. InceptionV3 Hyperparameters Optimization Comparison.

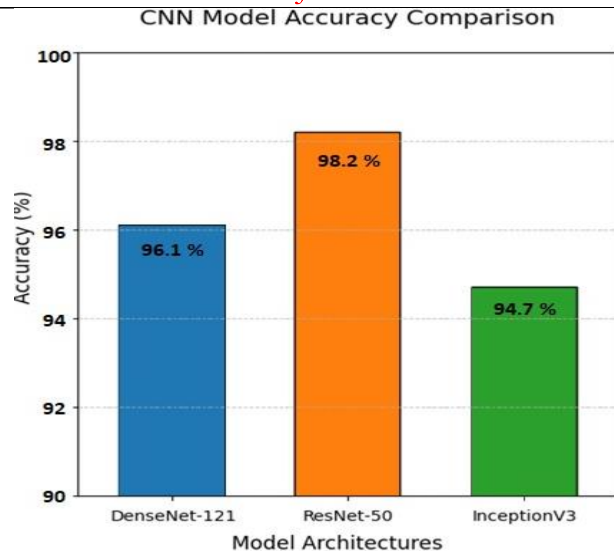

Figure 8. Comparative analysis of DenseNet-121, ResNet-50, and InceptionV3 CNN architecture.

Sample Standard Deviation:

The best performing models with optimal parameters were trained three times in order to check the model's stability, reliability, and generalizability. The given below is data for each model.

Table 7. DenseNet-121 test trainings for optimized parameters.

CNN Model	Training	Learning Rate	Batch Size	Optimizer	Accuracy
DenseNet-121	1	**1e-4**	**32**	**Adam**	**96.1%**
	2				**95.8%**
	3				**94.8%**

Table 8. ResNet-50 test trainings for optimized parameters

CNN Model	Training	Learning Rate	Batch Size	Optimizer	Accuracy
ResNet-50	1	**1e-4**	**32**	**Adam**	**98.2%**
	2				**98.0%**
	3				**97.7%**

Table 9. InceptionV3 test trainings for optimized parameters.

CNN Model	Training	Learning Rate	Batch Size	Optimizer	Accuracy
InceptionV3	1	**1e-4**	**32**	**RMSprop**	**94.7%**
	2				**91.4%**
	3				**84.3%**

Sample Standard Deviation for above mentioned CNN models is:
DenseNet-121: The sample standard deviation of 96.1, 95.8, and 94.8 is approximately: 0.681
ResNet-50: The sample standard deviation of 98.2, 98.0, and 97.7 is approximately: 0.252
InceptionV3: The sample standard deviation of 94.7, 91.4, and 84.3 is approximately: 5.31

Results Discussion:

The experimental results indicate that **ResNet-50** outperformed the other CNN architectures, achieving the highest classification accuracy for tomato leaf disease detection. The **sample standard deviation** provides insight into the stability and reliability of each model. Both ResNet-50 and DenseNet-121 demonstrated low variability across multiple training runs, reflecting consistent performance under repeated experiments. In contrast, InceptionV3 exhibited higher variability in accuracy, suggesting that the model with the current parameter settings requires further optimization to achieve stable results.

Conclusion:

The experiments demonstrate that **hyperparameter optimization** significantly enhances the performance of CNN models for plant disease classification. Among the evaluated architectures, **ResNet-50 achieved the best overall performance**, followed closely by DenseNet-121. InceptionV3, while showing improved results through tuning, still performed slightly lower than the other two models. The analysis of standard deviation further confirmed the **robustness and stability** of ResNet-50 and DenseNet-121 under the proposed hyperparameter configurations.

Optimal settings generally favored the **Adam optimizer**, moderate **learning rates of 1e-4**, and **smaller batch sizes of 32**, highlighting these factors as crucial for efficient training and high model accuracy. The findings emphasize the importance of systematic tuning of learning rate, batch size, and optimizer selection for enhancing deep learning performance.

These results provide practical insights for designing **CNN-based solutions** for automated plant disease detection and demonstrate the effectiveness of leveraging **cloud-based platforms like Google Colab** for scalable and accessible experimentation. Future research could explore additional CNN architectures, extended hyperparameter ranges, and more diverse datasets to validate and generalize these findings for real-world agricultural applications.

Author Contributions:

Syed Sohail Ahmed Shah: Conducted Research, Conceptualization, Methodology, Writing- Original draft preparation, implementation, Visualization. Prof. Dr. Mujeeb-u-Rehman Maree: Supervision, experiment design, reviewing, writing, and editing. Dr. Aftab Ahmed Chandio: Supervision, reviewing, writing, and editing. Dr. Mujeeb-ur-Rehman Jamali: reviewing and editing.

Compliance with Ethical Standards:

It is declared that all authors don't have any conflict of interest. It is also declared that this article does not contain any studies with human participants performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

Funding Information:

No funding is granted in this research work.

References:

[1] J. Sharma, "Optimizing Tomato Crop Health: Disease Detection with," *Proc. 5th Int. Conf. Smart Electron. Commun. ICOSEC 2024*, pp. 1259–1264, 2024, doi: 10.1109/ICOSEC61587.2024.10722724.

[2] S. A. A. Qadri, N. F. Huang, T. M. Wani, and S. A. Bhat, "Advances and Challenges in Computer Vision for Image-Based Plant Disease Detection: A Comprehensive Survey of Machine and Deep Learning Approaches," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 2639–2670, 2025, doi: 10.1109/TASE.2024.3382731.

[3] F. Bin Jashim, F. R. Refat, M. H. Karim, F. U. Mahmud, and M. I. H. Siddiqui, "Interpretable mango leaf disease detection using a hybrid CNN–transformer model with GLCM features," *Int. J. Sci. Res. Arch.*, vol. 15, no. 2, pp. 1518–1535, May 2025, doi: 10.30574/IJSRA.2025.15.2.1510.

[4] H. J. Runhua Zhang, "Optimization Methods, Challenges, and Opportunities for Edge Inference: A Comprehensive Survey," *Electronics*, vol. 14, no. 7, p. 1345, 2025, doi: https://doi.org/10.3390/electronics14071345.

[5] N. S. K. V. Krishna Pratap, "Revolutionizing Plant Disease Detection in Leaves: An Innovative Hybrid ABOCNN Framework for Advanced and Accurate

Identification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 2, 2024, doi: 10.14569/IJACSA.2024.0150226.
[6] S. M. R. A. M. Rajibul Islam, Asif Mahmod Tusher Siddique, Amiruzzaman, Abdullah-Al-Wadud, "An Efficient Technique for Recognizing Tomato Leaf Disease Based on the Most Effective Deep CNN Hyperparameters," *Ann. Emerg. Technol. Comput.*, vol. 7, no. 1, 2023, [Online]. Available: https://aetic.theiaer.org/archive/v7/v7n1/p1.html
[7] A. M. W. Hsing Chung Chen, "AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf," *Electronics*, vol. 11, no. 6, p. 951, 2022, doi: https://doi.org/10.3390/electronics11060951.
[8] Y. D. Radhika Bhagwat, "Comprehensive Multilayer Convolutional Neural Network for Plant Disease Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, 2021, doi: 10.14569/IJACSA.2021.0120125.
[9] S. Y. Iftikhar Ahmad, Muhammad Hamid, "Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection," *Complexity*, 2020, doi: https://doi.org/10.1155/2020/8812019.
[10] M. S. M. A. Reem Mohammed Jasim Al-Akkam, "Plants Leaf Diseases Detection Using Deep Learning," *Iraqi J. Sci.*, vol. 63, no. 2, 2022, [Online]. Available: https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/4064
[11] J. S. L. Lih Poh LIN, "Detection and Categorization of Tomato Leaf Diseases Using Deep Learning," *Int. J. Appl. Sci. Technol. Eng.*, vol. 1, no. 1, 2023, doi: https://doi.org/10.24912/ijaste.v1.i1.282-291.
[12] V. J. Supriya Shrivastav, "A Fusion Method for Detection and Classification of Diseases in Tomato Plants Using Swarm-based Deep Learning," *Int. J. Exp. Res. Rev.*, vol. 45, 2024, [Online]. Available: https://qtanalytics.in/journals/index.php/IJERR/article/view/3900
[13] K. B. Kamaldeep Kaur, "Enhancing Plant Disease Detection using Advanced Deep Learning Models," *Indian J. Sci. Technol.*, vol. 17, no. 17, pp. 1755–1766, 2024, doi: 10.17485/IJST/v17i17.536.
[14] C. Vengaiah and S. R. Konda, "Improving Tomato Leaf Disease Detection with DenseNet-121 Architecture," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 3, pp. 442–448, Jul. 2023, Accessed: Nov. 13, 2025. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/3194
[15] M. B. Suren Kumar Sahu, "Plant Disease Classification Using Densenet-121 Architecture And Flask," *JETIR*, vol. 10, no. 9, 2023, [Online]. Available: https://www.jetir.org/view?paper=JETIR2309022
[16] W. J. Jin Liang, "A ResNet50-DPA model for tomato leaf disease identification," *Front. Plant Sci.*, vol. 14, 2023, doi: https://doi.org/10.3389/fpls.2023.1258658.
[17] A. S. Laxmi Upadhyay, "Evaluation of Enhanced Resnet-50 Based Deep Learning Classifier for Tomato Leaf Disease Detection and Classification," *J. Electr. Syst.*, vol. 20, no. 3s, 2024, [Online]. Available: https://journal.esrgroups.org/jes/article/view/1856
[18] S. I. Jiraporn Thomkaew, "Improvement Classification Approach in Tomato Leaf Disease using Modified Visual Geometry Group (VGG)-InceptionV3," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, 2022, doi: 10.14569/IJACSA.2022.0131244.