JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 1 AUGUST 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i1.7014

COMPARATIVE ANALYSIS OF CNN ARCHITECTURES FOR TOMATO LEAF DISEASE CLASSIFICATION USING TRANSFER LEARNING

Anton[1,2*]; Supriadi Rustad[1]; Guruh Fajar Shidik[1]; Abdul Syukur[1]

Computer Science[1]
Universitas Dian Nuswantoro, Semarang, Indonesia[1]
https://dinus.ac.id/[1]
p41201900016@mhs.dinus.ac.id*; srustad@dsn.dinus.ac.id; guruh.fajar@research.dinus.ac.id; abdul_s@dosen.dinus.ac.id

Informatics[2]
Universitas Nusa Mandiri, Jakarta, Indonesia[2]
www.nusamandiri.ac.id[2]
anton@nusamandiri.ac.id*

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract*—Tomato is one of the widely available horticultural products and holds significant economic value in Indonesia. However, its productivity is often disrupted by various leaf diseases. This study aims to compare the performance of three CNN architectures—DenseNet121, Xception, and MobileNetV2—in classifying tomato leaf diseases. The dataset used consists of 10,000 balanced images across ten classes: Bacterial Spot, Septoria Leaf Spot, Early Blight, Late Blight, Mosaic Virus, Yellow Leaf Curl Virus, Leaf Mold, Target Spot, Spider Mites Two-Spotted Spider Mite, and Healthy. All images were resized to 224x224 pixels and divided into 80% training data and 20% test data. Augmentation techniques were applied to balance the data across classes. Experimental results show that the Xception architecture outperforms the other models, achieving an accuracy of 98.79%, with a precision of 98.80%, recall of 98.79%, and an F1-Score of 98.78%. These findings indicate that the Xception model is highly effective for plant disease classification and is suitable for implementation in environments with limited resources.*

Keywords*: CNN, DenseNet121, MobileNetV2, Tomato Leaf Disease, Xception*

Intisari—*Tomat merupakan salah satu produk hortikultura yang banyak tersedia dan memiliki nilai ekonomi yang signifikan di Indonesia. Namun, produktivitasnya seringkali terganggu oleh berbagai penyakit daun. Penelitian ini bertujuan untuk membandingkan kinerja tiga arsitektur CNN—DenseNet121, Xception, dan MobileNetV2—dalam mengklasifikasikan penyakit daun tomat. Dataset yang digunakan terdiri dari 10.000 citra seimbang yang terbagi dalam sepuluh kelas: Bercak Bakteri, Bercak Daun Septoria, Hawar Daun Dini, Hawar Daun Akhir, Virus Mosaik, Virus Keriting Daun Kuning, Jamur Daun, Bercak Target, Tungau Laba-laba Berbintik Dua, dan Sehat. Semua citra diubah ukurannya menjadi 224x224 piksel dan dibagi menjadi 80% data latih dan 20% data uji. Teknik augmentasi diterapkan untuk menyeimbangkan data di seluruh kelas. Hasil eksperimen menunjukkan bahwa arsitektur Xception mengungguli model-model lain, dengan akurasi 98,79%, presisi 98,80%, recall 98,79%, dan F1-Score 98,78%. Temuan ini menunjukkan bahwa model Xception sangat efektif untuk klasifikasi penyakit tanaman dan cocok untuk diimplementasikan di lingkungan dengan sumber daya terbatas.*

Kata Kunci*: CNN, DenseNet121, MobileNetV2, Penyakit Daun Tomat, Xception.*

INTRODUCTION

The agricultural sector has an influence on economic growth in the ASEAN region, including Indonesia [1]. One of the important agricultural commodities is tomatoes, which have significant economic value for farmers [2]. The agricultural sector is very challenged by the presence of plant pests and diseases [3]. However, tomato production is often hampered by various diseases that attack the leaves, causing economic losses for farmers [4]. Additionally, diseases can be categorized according to the crop portion that is impacted. The disease's symptoms may manifest on the leaf, panicle, stem, or sheath [5].

Early detection of tomato leaf disease is crucial to protect crops and increase crop yields [6]. Like other plants, tomato plants are susceptible to microorganism attacks that can cause disease [7]. Plant diseases that impact different plant sections, such as leaves, stems, and roots, are frequently caused by bacteria and fungi [8].

Therefore, to lessen the financial losses brought on by this illness, an effective control system must be developed [9]. Using information technology to deliver timely and pertinent agricultural information is one approach to solving this issue and assisting farmers in making decisions [10]. The identification of plant diseases was proposed by [11] using the method of identifying diseases of rice disease based on the Convolutional Neural Network (CNN) technique. The use of digital image processing can help in the early detection of leaf diseases in plants, such as in mango plants [12]. Smart techniques are slowly replacing crop monitoring because they are more reliable, accurate, fast, and economical [13]. CNNs have proven effective in automating disease detection in crops like tomatoes, allowing for quicker responses to potentially devastating pest outbreaks [14].

Observing the afflicted areas of the plant, particularly the leaves, which exhibit symptoms including color changes and the emergence of spots, can help identify tomato diseases early [6]. Creating precise picture classifications for plant disease detection necessitates a dataset containing validated photos of both healthy and afflicted plants [15].

In the realm of agricultural diagnostics, particularly concerning tomato leaf diseases, deep learning methodologies have proven highly effective. Several approaches based on convolutional neural networks (CNNs) and hybrid models have been employed to enhance the accuracy and efficiency of classifying tomato leaf diseases [16]. When compared to other algorithms, CNN has the advantage of producing superior categorization results. Compared to other Deep Learning models, CNN is preferred for image identification because of its high accuracy [17]. The CNN architecture comprises a convolutional layer, a pooling layer, and a fully connected layer, with the convolutional layer serving as the core component of its computations [18].

An effective approach for detecting and classifying tomato leaf diseases uses a CNN with two convolutional and pooling layers. This method outperformed pre-trained models such as InceptionV3, ResNet152, and VGG19, reaching 98% training accuracy and 88.17% testing accuracy [19]. A study on tomato plant disease classification using CNN architectures, namely AlexNet and VGG16, utilized 13,262 images from the PlantVillage dataset. The results showed that VGG16 achieved an accuracy of 97.29%, while AlexNet performed slightly better with 97.49% [20].

Compared to other algorithms, CNN offers better classification performance. It is often the top choice for image recognition among deep learning models due to its high accuracy [21]. The CNN architecture consists of three main parts: a convolutional layer, a pooling layer, and a fully connected layer. At its core, the convolutional layer is the key element that drives CNN’s computations [22].

Research [23]classified tomato leaf diseases using a CNN model with DenseNet architecture. 12,332 photos with a resolution of 250 by 250 pixels that were split up into four illness classifications made up the dataset that was used. 90% of the data was used for training, and 10% was used for validation. After preprocessing—which included augmenting the training data—the model was trained using the DenseNet architecture, optimized to reach 98.06% accuracy with an average training time of 3 minutes per epoch. Another Research [24] used data from 15,408 photos in 4 classes to propose pre-trained models of DenseNet and EfficientNetB0. Eighty percent of the dataset was allocated for training, and the remaining twenty percent was reserved for testing. The accuracy values of the suggested DenseNet and EfficientNetB0 models were 96.26% and 97.91%, respectively. The goal of the study [25] was to use a transfer learning technique to identify tomato leaf mold using DenseNet121. Transfer learning applies a pre-trained model to a new task, avoiding training from scratch [26]. With 2,283 photos from three categories (leaf mold, healthy, and others), the model's accuracy, precision, and recall were 92.6%, 93.3%, and 93%, respectively. Although the model hasn't been tested on actual field photos yet, transfer learning worked well with expedited training and minimal data.

MobileNetV2, a lightweight deep learning model, is widely used in agriculture for efficient tomato disease detection due to its low computational demands and high accuracy. It has achieved up to 99.11% accuracy in classifying tomato leaf diseases, demonstrating its effectiveness for real-time applications [27]. MobileNetV2 is considered suitable for resource-constrained devices, although the study did not account for variations in lighting and background conditions. For instance, the MobileNetV2, a model similar in efficiency to Xception, has been used to optimize the detection of tomato leaf diseases, classifying seven disease categories with a notable accuracy of 98% [28].

Hybrid models combining CNNs with transformers are particularly effective, achieving high accuracy rates, such as 99.45% on the PlantVillage dataset, by harnessing both local feature extraction and global dependency capturing capabilities [29]

Using preprocessing, SMOTE, and CNN, this study created a system that can automatically find apple leaf diseases. The proposed model was more accurate than DenseNet121 (95%), InceptionV3 (92%), and InceptionResNetV2 (91%) [30]. Research [31] proposes a transfer learning-based CNN model (InceptionV3, VGG16, ResNet, etc.) for detecting biotic diseases in rice leaves. Through preprocessing, semantic segmentation, and DNN classification, this model achieves 96.4% accuracy and outperforms other models.

Many studies have used CNN architectures to classify plant diseases; however, there haven't been many comparisons using balanced and preprocessed multi-class tomato leaf disease datasets. Also, lightweight architectures like MobileNetV2 haven't been compared very much in limited resources. This study fills these gaps by testing DenseNet121, Xception, and MobileNetV2 on a standardized dataset to find the best architectures for quickly and accurately classifying tomato leaf diseases. The DenseneNet121, MobileNetV2, and Xception architecture models developed in this study will help tomato growers identify diseased tomato plants. This study uses the same dataset and preprocessing procedure to examine how well the three models perform in classifying tomato plant illnesses. This study makes two contributions. First, especially when used with picture datasets, the created model may be a solution for identifying tomato plant illnesses. Second, the findings of this study's model performance comparison can serve as a guide for future research into creating a model for classifying plant diseases based on images.

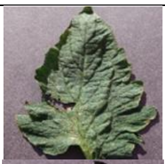
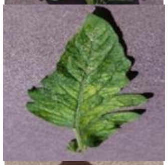
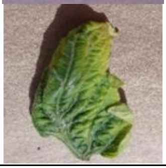
MATERIALS AND METHODS

Dataset

This study used the PlantVillage dataset from GitHub, containing over 54,000 images in 38 classes, widely used for training CNNs and hybrid models in plant disease classification [14]. We processed it using Google Colab and the Python programming language. This dataset contains 18,160 images across ten disease classes, sourced directly from the PlantVillage repository. While widely used in plant disease studies, the labels are based on the original dataset and have not been further verified by domain experts in this study, expert validation by agricultural specialists is planned for future phases. Table 1 summarizes the image distribution per class.

Table 1. Dataset of tomato plant leaves

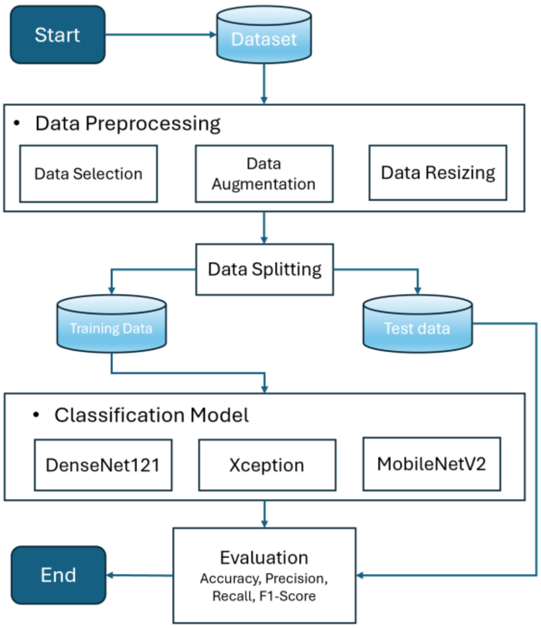
No	Class	amount of images	images
1	Bacterial spot	2127	
2	Early blight	1000	
3	Healthy	1591	
4	Late Blight	1909	
5	Leaf mold	952	
6	Septoria leaf spot	1771	
7	Spider Mites Two Spotted spider mite	1676	

No	Class	amount of images	images
8	Target Spot	1404	
9	Mosaic Virus	373	
10	Yellow leaf Curl Virus	5357	

Source: (Research Results, 2025)

Research Flow

The research began with the collection of a dataset consisting of images of plant leaves in various conditions, including healthy leaves and those infected by different diseases. The data selection process was carried out to choose a subset of images so that each class had an equal number of images. For the classes, Bacterial Spot, Early Blight, Mosaic Virus, Septoria Leaf Spot, Late Blight, Leaf Mold, Spider Mites (Two-Spotted Spider Mite), Yellow Leaf Curl Virus, Target Spot, and Healthy, 1,000 images from each category were selected for the next stage of the process. The research workflow is illustrated in the following flowchart.



Source: (Research Results, 2025)

Figure 1. Research flowchart

Preprocessing

The dataset underwent preprocessing, which involved three main steps: data selection to ensure only relevant and high-quality images were used for training; data augmentation [32], where techniques such as rotation, flipping, zooming, and shifting were applied to enrich data variation and improve model generalization [33]; and data resizing [34], where all images were resized to match the input dimensions required by the chosen CNN architectures. Image resizing was performed to ensure uniform resolution across all images. Variations in image size can affect the performance of the trained model [35]. The image resolution was standardized to 224 × 224 pixels for all classes used in the training process [36].

Data Splitting

After preprocessing, the dataset was divided into training and test subsets. The training data is distributed as 80% and the testing data as 20% of the total available data [37], ensuring that the model's evaluation was performed on unseen data.

Classification Model

Three pretrained CNN architectures—DenseNet121, Xception, and MobileNetV2—were utilized for classification tasks. Each model was trained on the training set and evaluated on the test set.

Evaluation

Evaluation metrics included accuracy, precision, recall, and F1-score to assess performance comprehensively. A comparative analysis was then carried out to identify the best-performing architecture for leaf disease classification, with the model scoring highest across the evaluation metrics deemed the most optimal for the task.

Table 1. Characteristics of the system used for experimentation.

Hardware/Software	Characteristics
Number of CPU's	4
GPU's Feature	NVIDIA T4
GPU RAM	15 Gb
System RAM	51Gb
Disk	235,7 Gb
Number of cuda cores	2560
Number of tensor cores	320

Source: (Research Results, 2025)

DenseNet121

DenseNet121 is one of the CNN architectures commonly used for image classification. DenseNet121 is known for its densely connected layers, which improve the flow of information and

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

gradients throughout the network, addressing the vanishing gradient problem [38] According to a study, DenseNet121 demonstrated competitive performance when compared to other architectures such as visual transformers in disease classification tasks. Although visual transformers slightly surpass it in AUC for certain tasks, it remains highly efficient in handling sample data and demonstrates strong resilience to hidden stratification [39]. This architecture also excels in scenarios requiring feature reuse, which enhances classification rates under various conditions while maintaining low computational costs, as evidenced in underwater target recognition tasks [40].

Table 2. Configuration of the DenseNet121 Architecture

Layer Type	Filters / Units	Kernel Size	Stride	Activation
Conv	64	7×7	2	ReLU
Max Pooling	-	3×3	2	-
Dense Block 1	6 × (32 filters)	1×1 & 3×3	1	ReLU
Transition Layer	128	1×1, 2×2 pool	1, 2	ReLU
Dense Block 2	12 × (32 filters)	1×1 & 3×3	1	ReLU
Transition Layer	256	1×1, 2×2 pool	1, 2	ReLU
Dense Block 3	24 × (32 filters)	1×1 & 3×3	1	ReLU
Transition Layer	512	1×1, 2×2 pool	1, 2	ReLU
Dense Block 4	16 × (32 filters)	1×1 & 3×3	1	ReLU
Global Avg Pool	-	-	-	-
Fully Connected	1000 (ImageNet)	-	-	Softmax

Source: (S. Anwar, 2023 [41])

Xception

Xception or Extreme Inception is an architecture developed by Francois Chollet, a radical evolution of the Inception architecture that adopts a thorough depthwise separable convolution approach, with modular structures and residual connections to improve feature processing efficiency [42].

Table 3. Configuration of the Xception Architecture

Layer Type	Filters / Units	Kernel Size	Stride	Activation
Entry Flow Conv	32	3×3	2	ReLU
Entry Flow Conv	64	3×3	1	ReLU
Entry Flow Block 1	128	Separable 3×3	2	ReLU
Entry Flow Block 2	256	Separable 3×3	2	ReLU
Entry Flow Block 3	728	Separable 3×3	2	ReLU
Middle Flow (8×)	728	Separable 3×3	1	ReLU
Exit Flow Block 1	1024	Separable 3×3	2	ReLU
Exit Flow Conv	1536	Separable 3×3	1	ReLU
Exit Flow Conv	2048	Separable 3×3	1	ReLU
Global Avg Pool	-	-	-	-
Fully Connected	1000 (Image Net)	-	-	Softmax

Source: (H. Ulutaş and V. Aslantaş, 2023 [43])

MobileNetV2

MobileNetV2 is designed for mobile and resource-constrained environments, emphasizing efficiency and speed. It uses depthwise separable convolutions, greatly reducing computational costs and model size without sacrificing accuracy. While not explicitly mentioned in the provided studies, MobileNetV2's architecture is generally praised for its quick inference and adaptability in scenarios requiring efficient computations, such as running on low-power devices like Raspberry Pi [44].

Table 4. Configuration of the MobileNetV2 Architecture

Layer Type	Output Channels	Kernel Size	Stride	Expansion Factor (t)	Activation
Conv2D	32	3×3	2	-	ReLU6
Bottleneck (1)	16	3×3	1	1	Linear
Bottleneck (2–3)	24	3×3	1-Feb	6	ReLU6
Bottleneck (4–6)	32	3×3	1-Feb	6	ReLU6
Bottleneck (7–10)	64	3×3	1-Feb	6	ReLU6
Bottleneck (11–13)	96	3×3	1	6	ReLU6
Bottleneck (14–16)	160	3×3	1-Feb	6	ReLU6
Bottleneck (17)	320	3×3	1	6	ReLU6
Conv2D	1280	1×1	1	-	ReLU6

Layer Type	Output Channels	Kernel Size	Stride	Expansion Factor (t)	Activation
Global Avg Pool	-	-	-	-	-
Fully Connected	1000 (ImageNet)	-	-	-	Softmax

Source: (G. Mukherjee, A. Chatterjee, and B. Tudu, 2022 [28])

Despite having distinct traits and advantages, all three architectures have demonstrated efficacy in a variety of image recognition applications. To determine the most suitable architecture for image classification, such as identifying plant leaf diseases, it is essential to compare the performance of models like MobileNetV2, Xception, and DenseNet121.

This study assesses the classification model's performance using a confusion matrix [45]. The anticipated value and the actual value, displayed as a table, are described by the confusion matrix.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

$$F1-Score = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \qquad (4)$$

RESULTS AND DISCUSSION

The first stage of this study is data selection from the original dataset. The same number of images is chosen for each class to achieve a balanced distribution across all classes. The final image count after this process is presented in the table below.

Table 5. Data Selection Results

Code	Class	amount of images
C1	Bacterial spot	2127
C2	Early blight	1000
C3	Healthy	1591
C4	Late Blight	1909
C5	Leaf mold	952
C6	Septoria leaf spot	1771
C7	Spider Mites Two Spotted spider mite	1676
C8	Target Spot	1404
C9	Mosaic Virus	373
C10	Yellow leaf Curl Virus	5357

Source: (Research Results, 2025)

Furthermore, data augmentation was applied specifically to the *Mosaic Virus* and *Leaf Mold* classes, which initially contained 373 and 952 images, respectively. The augmentation process involved rotating the images within a range of -25 to 25 degrees and applying horizontal flipping with a probability of 0.5. As a result, 627 additional images were generated for the *Mosaic Virus* class, bringing its total to 1,000 images. Similarly, 48 new images were added to the *Leaf Mold* class, also increasing its total to 1,000 images. Table 6 presents the number of images per class after the augmentation process.

Table 6. Data Augmentation Results

Code	Class	A mount of images
C1	Bacterial spot	1000
C2	Early blight	1000
C3	Healthy	1000
C4	Late Blight	1000
C5	Leaf mold	1000
C6	Septoria leaf spot	1000
C7	Spider Mites Two Spotted spider mite	1000
C8	Target Spot	1000
C9	Mosaic Virus	1000
C10	Yellow leaf Curl Virus	1000

Source: (Research Results, 2025)

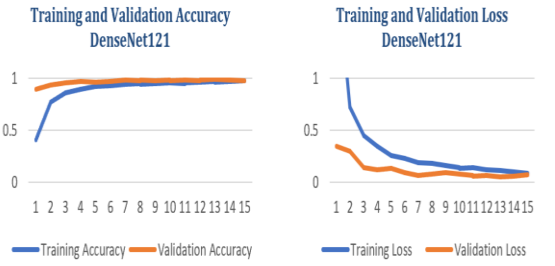
After data augmentation is done on the Mosaic Virus class, and the Leaf Mold class each has 1000 images. Once the ten classes are balanced in data quantity, image resizing is performed to standardize all image dimensions. This uniform size facilitates the convolution and pooling operations during training. In this study, all images were consistently resized to 224 x 224 pixels.

The next step is splitting the dataset into training and test sets. This study used 80% of the data to train the CNN model, while the remaining 20% was reserved for testing. The distribution of training and testing data is shown in the table below.

Each class, Bacterial Spot, Early Blight, Healthy, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites (Two-Spotted Spider Mite), Target Spot, Mosaic Virus, and Yellow Leaf Curl Virus, contains 800 images for training and 200 images for testing.

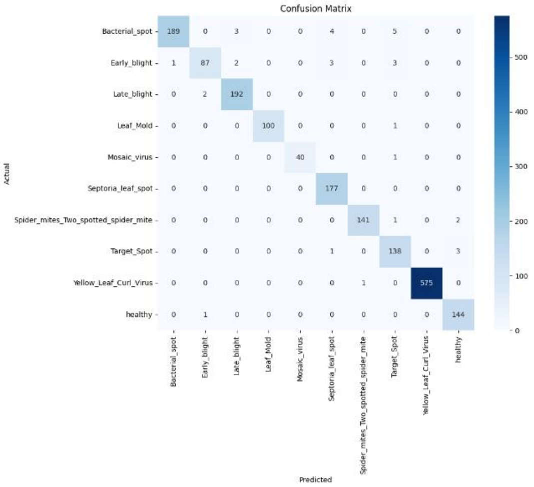
DenseNet121

The first model in this study applies the DenseNet121 architecture for dataset classification. Its evaluation results are presented in the figure and confusion matrix below.



Source: (Research Results, 2025)

Figure 2. Grafic Accuracy and Loss of DenseNet121



Source: (Research Results, 2025)

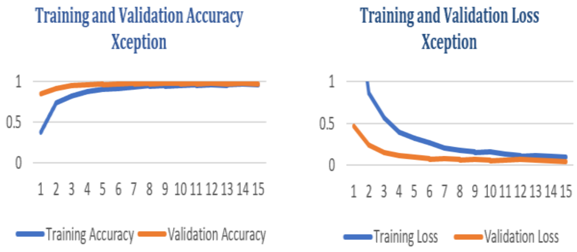
Figure 3. Confusion Matrix DenseNet121

The confusion between Early Blight and Septoria Leaf Spot, and between Spider Mites and Healthy leaves, may be attributed to similar visual features such as chlorotic patterns and small speckled lesions. These visual similarities likely challenge the feature extraction process of the model. Implementing feature enhancement techniques or class-specific augmentation could help reduce these misclassifications.

The DenseNet121 model exhibits exceptional learning capabilities, starting with a remarkable 119% validation accuracy surge from epoch 1 (89.10%) to epoch 7 (98.51%), while training accuracy simultaneously grew 132% (40.67% → 94.41%). After this rapid convergence phase, the model maintained outstanding stability with validation accuracy consistently above 97.41% for the remaining epochs, peaking at 98.79% (epoch 13) with a minimal validation loss of 0.0470. Although training accuracy continued climbing to 97.42% by epoch 15, slight validation fluctuations occurred post-peak (98.79% → 98.18%), possibly indicating minor overfitting despite an overall 96% reduction in training loss (2.1158 → 0.0834). The highly efficient training process maintained consistent ~205ms/step speeds after the initial epoch, demonstrating robust hardware utilization throughout the 15-epoch cycle.

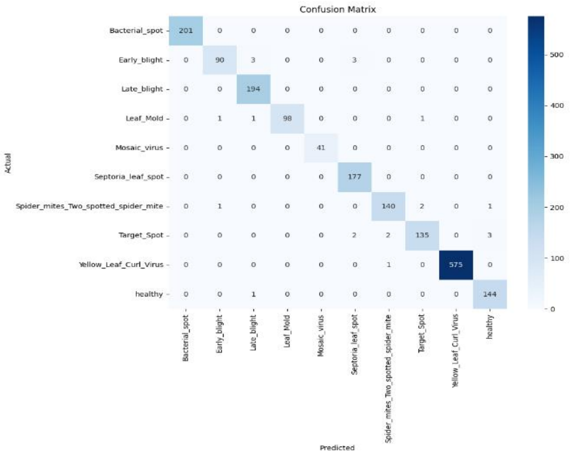
Xception

The second model uses the Xception architecture. The results of the evaluation are shown in the following garfic and confusion matrix:



Source: (Research Results, 2025)

Figure 4. Graphic Accuracy and Loss of Xception



Source: (Research Results, 2025)

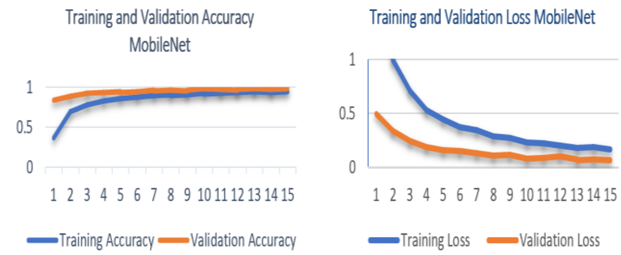
Figure 5. Confusion Matrix Xception

The confusion matrix shows certain misclassification patterns, especially between Early Blight and Septoria Leaf Spot and between Spider Mites and healthy leaves. These mistakes are probably caused by similar visual elements, including chlorotic patches, necrotic areas, or textural patterns that make it hard for the model to tell them apart. There are also fewer mistakes when describing the difference between Late Blight and Early Blight, as well as Target Spot and Septoria Leaf Spot. This could be because the lesions look or are the same. These results show that it may be hard to classify diseases accurately because they have comparable features. In the future, improvements could include targeted augmentation, feature engineering specific to a domain, or better combining attention-based processes to tell apart classes that look comparable.

The Xception model demonstrates rapid initial convergence with validation accuracy surging from 84.69% to 97.25% within just five epochs, while training accuracy simultaneously increased 139% (37.79% → 90.29%) and training loss decreased 85% (2.2043 → 0.3265). During subsequent epochs, the model exhibited exceptional stability with validation accuracy consistently above 97.25% and gradually climbing to its peak of 98.18% at epoch 14 (loss 0.0541), supported by steady training accuracy improvements reaching 97.10% by epoch 14. Though slight fluctuations occurred in the final epoch (validation accuracy 98.07%), the model maintained remarkably consistent processing speeds averaging 225ms/step after epoch 1, ultimately achieving a 157% cumulative training accuracy gain (37.79% → 96.96%) and 95.5% training loss reduction (2.2043 → 0.0987) across the 15-epoch cycle with minimal validation volatility (±0.6% after epoch 5).

MobileNet

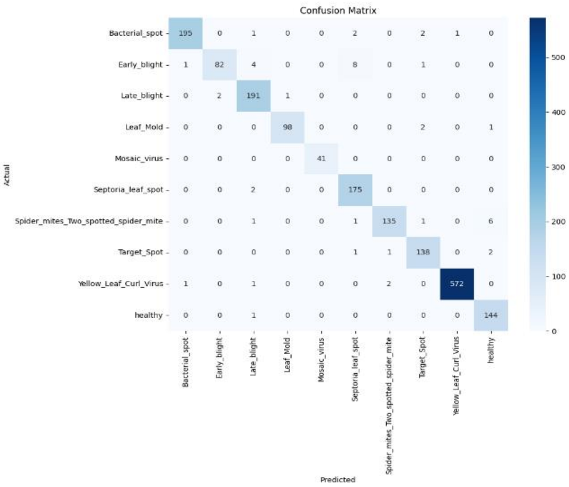
The third model applies the MobileNetV2 architecture, with its evaluation results presented in the following graph and confusion matrix.



Source: (Research Results, 2025)

Figure 6. Grafic Accuracy and Loss of mobileNet

The model demonstrates rapid and stable convergence over 15 epochs, with training accuracy surging from 36.60% to 94.61% and validation accuracy leaping from 83.81% to 98.18%. Most significant improvements occurred in the first 7 epochs where training accuracy increased by 142% (36.6% → 89.5%) and validation loss dropped 74% (0.4913 → 0.1288). After epoch 7, the model entered a refinement phase with slower but consistent gains, ultimately achieving peak validation performance at epoch 15 (98.18% accuracy, 0.0685 loss) while maintaining minimal training-validation gaps (3.57% accuracy difference). The dramatic time-per-step reduction from 13s to ≈200ms after epoch 1 indicates efficient hardware utilization, and steadily decreasing losses without rebound patterns confirm effective learning without overfitting throughout the training cycle.



	Bacterial_spot	Early_blight	Late_blight	Leaf_Mold	Mosaic_virus	Septoria_leaf_spot	Spider_mites_Two_spotted_spider_mite	Target_Spot	Yellow_Leaf_Curl_Virus	healthy
Bacterial_spot	195	0	1	0	0	2	0	2	1	0
Early_blight	1	82	4	0	0	8	0	1	0	0
Late_blight	0	2	191	1	0	0	0	0	0	0
Leaf_Mold	0	0	0	98	0	0	0	2	0	1
Mosaic_virus	0	0	0	0	41	0	0	0	0	0
Septoria_leaf_spot	0	0	2	0	0	175	0	0	0	0
Spider_mites_Two_spotted_spider_mite	0	0	1	0	0	1	135	1	0	6
Target_Spot	0	0	0	0	0	1	1	138	0	2
Yellow_Leaf_Curl_Virus	1	0	1	0	0	0	2	0	572	0
healthy	0	0	1	0	0	0	0	0	0	144

Source: (Research Results, 2025)

Figure 7. Confusion Matrix MobileNet

The confusion matrix reveals that the classifications are mostly correct, but some are wrong in Early Blight, which is commonly confused with Late Blight, Leaf Mold, or Target Spot because they look similar. Sometimes, spider mites are called "healthy," which may be due to how the leaves look. Bacterial Spot's small mistakes also point to feature overlap. Adding attention mechanisms or improving class-specific features may help clear up these misunderstandings.

Despite the models' high training and validation accuracy, minor variations in validation accuracy, especially in DenseNet121, point to a possible overfitting risk. Dropout regularization, data augmentation expansion, early halting, and k-fold cross-validation are possible enhancements that could increase generalization.

The MobileNet confusion matrix reveals strong overall performance with distinct error patterns: Mosaic_virus achieved perfect classification (41/41 correct), while Septoria_leaf_spot (175/177) and Yellow Leaf Curl Virus (572/576) demonstrated near-flawless recognition despite the latter's large sample size. However, Early blight showed significant misclassification challenges with only 82/96 correct predictions—8 confused with Septoria leaf spot and 4 with Late blight. Bacterial spot (195/201) exhibited minor confusion with Septoria leaf spot (2 cases) and Target Spot (2 cases), whereas Spider mites experienced notable misidentification as healthy leaves (6/144 errors). Leaf Mold (98/101) primarily confused Target Spot cases (2 errors), and Target Spot itself showed inconsistencies requiring verification between predicted (138) and expected (142) samples. The matrix highlights particular difficulty distinguishing visually similar diseases like Early blight/Septoria

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

and Spider mites/healthy foliage, suggesting feature extraction enhancements for these classes.

Model Performance Comparison

We compared the results of each tested categorization model to see how well they worked. The table below compares the three CNN architectures regarding accuracy, precision, recall, and F1-score.

Table 10. Model Evaluation Comparison

Model	Accuracy	Precision	Recall	F1-Score
MobileNet	0.9747	0.9753	0.9747	0.9745
DenseNet121	0.9813	0.9817	0.9813	0.9813
Xception	0.9879	0.988	0.9879	0.9878

Source: (Research Results, 2025)

The Xception model recorded the highest performance, achieving 98.79% accuracy, 98.80% precision, 98.79% recall, and a 98.78% F1-score. These results indicate that the model is highly consistent in recognizing and classifying various types of tomato leaf diseases with a very low error rate. The high F1-Score also demonstrates a balance between precision and recall, meaning the model is not only accurate but also reliable in detecting all disease classes. Meanwhile, DenseNet121 ranked second with an accuracy of 98.13%, precision of 98.17%, recall of 98.13%, and F1-Score of 98.13%. This model shows very competitive performance and is close to that of Xception, although slightly lower in terms of precision and generalization. The MobileNet model, while more lightweight and computationally efficient, performed well with an accuracy of 97.47%, precision of 97.53%, recall of 97.47%, and F1-Score of 97.45%. Although slightly trailing the other two models in performance, MobileNet remains a strong choice for deployment on resource-limited devices thanks to its low architectural complexity.

CONCLUSION

Using DenseNet121, Xception, and MobileNet architectures to detect tomato leaf diseases can produce strong results. Their performance can be further improved by increasing both the input image size and the amount of training data. Xception generally delivers the best performance on the given dataset, thanks to its more complex layer structure than DenseNet121 and MobileNet. The Xception model achieved the highest accuracy of 98.79%, with a precision of 98.80%, recall of 98.79%, and an F1-Score of 98.78%. These values are better than those of DenseNet121 and MobileNet. A limitation of this study is that only 10,000 images were used, and only three architectures were employed—DenseNet121, MobileNet, and Xception. Future research could compare the performance of other CNN architectures—such as VGGNet, DenseNet, ResNet, and EfficientNet—using similar datasets but with larger images. This study had some good findings, but only tested three CNN architectures on a dataset of images that were all the same size. Also, they didn't talk about how things can change in the actual world, such as how the sunlight changes or how leaves block the view. Future work should look at bigger datasets, get validation from domain experts, and look into deploying on edge devices. Lightweight models like MobileNetV2 are good for mobile or embedded systems used in farming since they are easy to operate.

REFERENCE

[1] Inayah Rahman, Andini Cahyaning Pratiwi, and Putri Rizka Citaningati, "The Role Of The Agricultural Sector In Economic Growth In The Asean Region," *Jurnal Ekonomi Pertanian dan Agribisnis (JEPA)*, vol. 8, no. 1, 2024.

[2] BPS-RI, *Statistik Hortikultura 2020*, vol. 2020, no. 30. 2020. doi: 10.21608/jkom.2020.120332.

[3] N. Khasawneh, E. Faouri, and M. Fraiwan, "Automatic Detection of Tomato Diseases Using Deep Transfer Learning," *Applied Sciences*, 2022, doi: 10.3390/app12178467.

[4] Wahyu Nurwijayo, "Penyakit Tanaman Tomat: Jenis, Penyebab, Gejala, dan Cara Penanggulangan," https://gdm.id/penyakit-tanaman-tomat/. Accessed: Aug. 15, 2025. [Online]. Available: https://gdm.id/penyakit-tanaman-tomat/

[5] M. Dutta *et al.*, "Monitoring Root and Shoot Characteristics for the Sustainable Growth of Barley Using an IoT-Enabled Hydroponic System and AquaCrop Simulator," *Sustainability*, vol. 15, no. 5, 2023, doi: 10.3390/su15054396.

[6] D. L. Shanthi, K. Vinutha, N. Ashwini, and S. Vashistha, "Tomato Leaf Disease Detection Using CNN," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2975–2984. doi: 10.1016/j.procs.2024.04.281.

[7] A. Jasani, M. Dholi, and S. Purkar, "Tomato Leaf Disease Detection," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 5, pp. 918–922, May 2022, doi: 10.22214/ijraset.2022.41918.

[8] P. Agarwal, S. Saha, and P. Hariprasad, "Agro-Industrial-Residues as Potting Media: Physicochemical and Biological Characters and Their Influence on Plant Growth," *Biomass Convers Biorefin*, 2021, doi: 10.1007/s13399-021-01998-6.

[9] S. Sagar, M. Javed, and D. S. Doermann, "Leaf-Based Plant Disease Detection and Explainable AI," Dec. 2023, [Online]. Available: http://arxiv.org/abs/2404.16833

[10] Md. J. A. Soeb *et al.*, "Tea Leaf Disease Detection and Identification Based on YOLOv7 (YOLO-T)," *Sci Rep*, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-33270-4.

[11] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach," *Electronics (Basel)*, 2021, doi: 10.3390/electronics10121388.

[12] A. Rajbongshi, T. Khan, Md. M. Rahman, A. Pramanik, S. Md. Tanvir Siddiquee, and N. R. Chakraborty, "Recognition of Mango Leaf Disease Using Convolutional Neural Network Models: A Transfer Learning Approach," *Indonesian Journal of Electrical Engineering and Computer Science*, 2021, doi: 10.11591/ijeecs.v23.i3.pp1681-1688.

[13] A. S. A. N. D. E. Z. A. N. D. R. S. A. N. D. D. T. A. N. D. M. Fossa, "Smart Sensors and Smart Data for Precision Agriculture: A Review," Jul. 2024, *Multidisciplinary Digital Publishing Institute*. doi: 10.3390/s24082647.

[14] G. S. Hukkeri, B. C. Soundarya, H. L. Gururaj, and V. Ravi, "Classification of Various Plant Leaf Disease Using Pretrained Convolutional Neural Network On Imagenet," *Open Agric J*, vol. 18, no. 1, May 2024, doi: 10.2174/0118743315305194240408034912.

[15] S. Bensaadi and A. Louchene, "Low-Cost Convolutional Neural Network for Tomato Plant Diseases Classifiation," *Iaes International Journal of Artificial Intelligence (Ij-Ai)*, vol. 12, no. 1, p. 162, 2023, doi: 10.11591/ijai.v12.i1.pp162-170.

[16] M. Altalak, M. A. Uddin, A. Alajmi, and A. Rizg, "A Hybrid Approach for the Detection and Classification of Tomato Leaf Diseases," *Applied Sciences (Switzerland)*, vol. 12, no. 16, Aug. 2022, doi: 10.3390/app12168182.

[17] C. Sarkar, D. Gupta, U. Gupta, and B. B. Hazarika, "Leaf disease detection using machine learning and deep learning: Review and challenges," Sep. 01, 2023, *Elsevier Ltd*. doi: 10.1016/j.asoc.2023.110534.

[18] Y. Liu, G. Gao, and Z. Zhang, "Plant disease detection based on lightweight CNN model," in *2021 4th International Conference on Information and Computer Technologies (ICICT)*, 2021, pp. 64–68. doi: 10.1109/ICICT52872.2021.00018.

[19] G. Sakkarvarthi, G. W. Sathianesan, V. S. Murugan, A. J. Reddy, P. Jayagopal, and M. Elsisi, "Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network," *Electronics (Basel)*, vol. 11, no. 21, 2022, doi: 10.3390/electronics11213618.

[20] S. A. Wagle, R. Harikrishnan, S. H. Md Ali, and M. Faseehuddin, "Classification of Plant Leaves Using New Compact Convolutional Neural Network Models," *Plants*, 2021, doi: 10.3390/plants11010024.

[21] H. Chen *et al.*, "AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf," *Electronics (Basel)*, vol. 11, no. 6, p. 951, 2022, doi: 10.3390/electronics11060951.

[22] O. Elsherbiny, A. Elaraby, M. D. Alahmadi, M. Hamdan, and J. Gao, "Rapid Grapevine Health Diagnosis Based on Digital Imaging and Deep Learning," *Plants*, 2024, doi: 10.3390/plants13010135.

[23] M. Bakr, S. AbdelGaber, M. Nasr, and M. Hazman, "Tomato Disease Detection Model Based on Densenet and Transfer Learning," *Applied Computer Science*, vol. 18, no. 2, pp. 56–70, 2022, doi: 10.35784/acs-2022-13.

[24] H. Amin, A. Darwish, A. E. Hassanien, and M. Soliman, "End-to-End Deep Learning Model for Corn Leaf Disease Classification," *IEEE Access*, vol. 10, pp. 31103–31115, 2022, doi: 10.1109/ACCESS.2022.3159678.

[25] N. Awalia and A. Primajaya, "Identifikasi Penyakit Leaf Mold Daun Tomat Menggunakan Model Densenet121 Berbasis Transfer Learning," vol. 8, no. 1, 2022, [Online]. Available: http://ejournal.fikom-unasman.ac.id

[26] S. C. Huang and T. H. Le, *Principles and Labs for Deep Learning*. Elsevier, 2021. doi: 10.1016/B978-0-323-90198-7.09993-6.

[27] A. Mimi *et al.*, "Identifying Selected Diseases of Leaves using Deep Learning and Transfer Learning Models," *Machine Graphics and Vision*, vol. 32, no. 1, pp. 55–71, 2023, doi: 10.22630/MGV.2023.32.1.3.

[28] G. Mukherjee, A. Chatterjee, and B. Tudu, "Identification of the types of disease for tomato plants using a modified gray wolf optimization optimized MobileNetV2 convolutional neural network architecture

driven computer vision framework," *Concurr Comput*, vol. 34, no. 22, p. e7161, 2022, doi: https://doi.org/10.1002/cpe.7161.

[29] Z. Chen, X. Zhang, T. Lv, and G. Wang, "Using a Hybrid Convolutional Neural Network with a Transformer Model for Tomato Leaf Disease Detection," *Agronomy*, vol. 14, no. 4, 2024.

[30] H. ÇETİNER, "CLASSIFICATION OF APPLE LEAF DISEASES USING THE PROPOSED CONVOLUTION NEURAL NETWORK APPROACH," *Mühendislik Bilimleri ve Tasarım Dergisi*, vol. 9, no. 4, pp. 1130–1140, Dec. 2021, doi: 10.21923/jesd.980629.

[31] V. Gautam *et al.*, "A Transfer Learning-Based Artificial Intelligence Model for Leaf Disease Assessment," *Sustainability*, vol. 14, no. 20, p. 13610, 2022, doi: 10.3390/su142013610.

[32] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image Data Augmentation for Deep Learning: A Survey," Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.08610

[33] M. V. Shewale and R. D. Daruwala, "High performance deep learning architecture for early detection and classification of plant leaf disease," *J Agric Food Res*, vol. 14, Dec. 2023, doi: 10.1016/j.jafr.2023.100675.

[34] Tejaswini, P. Rastogi, S. Dua, Manikanta, and V. Dagar, "Early Disease Detection in Plants using CNN," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 3468–3478. doi: 10.1016/j.procs.2024.04.327.

[35] M. M. Mashamba, A. Telukdarie, I. Munien, U. Onkonkwo, and A. Vermeulen, "Detection of bacterial spot disease on tomato leaves using a Convolutional Neural Network (CNN)," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 602–609. doi: 10.1016/j.procs.2024.05.145.

[36] T. Sanida, A. Sideris, M. V. Sanida, and M. Dasygenis, "Tomato leaf disease identification via two–stage transfer learning approach," *Smart Agricultural Technology*, vol. 5, Oct. 2023, doi: 10.1016/j.atech.2023.100275.

[37] S. M. Kiran and D. N. Chandrappa, "Plant Leaf Disease Detection Using Efficient Image Processing and Machine Learning Algorithms," *Journal of Robotics and Control (JRC)*, vol. 4, no. 6, pp. 840–848, 2023, doi: 10.18196/jrc.v4i6.20342.

[38] M. Gogoi, V. Kumar, S. A. Begum, N. Sharma, and S. Kant, "Classification and Detection of Rice Diseases Using a 3-Stage CNN Architecture With Transfer Learning Approach," *Agriculture*, 2023, doi: 10.3390/agriculture13081505.

[39] Zachary R. Murphy, Kesavan Venkatesh, Jeremias Sulam, and Paul H. Yi, "Visual Transformers and Convolutional Neural Networks for Disease Classification on Radiographs: A Comparison of Performance, Sample Efficiency, and Hidden Stratification," *Radiol Artif Intell*, 2022.

[40] V.-S. Doan, T. Huynh-The, and D.-S. Kim, "Underwater Acoustic Target Classification Based on Dense Convolutional Neural Network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3029584.

[41] S. Anwar, S. R. Soomro, S. K. Baloch, A. A. Patoli, and A. R. Kolachi, "Performance Analysis of Deep Transfer Learning Models for the Automated Detection of Cotton Plant Diseases," *Engineering Technology & Applied Science Research*, 2023, doi: 10.48084/etasr.6187.

[42] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, and A. A. Khan, "Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network," *Ieee Access*, vol. 11, pp. 6594–6609, 2023, doi: 10.1109/access.2022.3232917.

[43] H. Ulutaş and V. Aslantaş, "Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model," *Electronics (Basel)*, vol. 12, no. 4, p. 827, 2023, doi: 10.3390/electronics12040827.

[44] R.-Y. Ju, T.-Y. Lin, J.-H. Jian, and J.-S. Chiang, "Efficient convolutional neural networks on Raspberry Pi for image classification," *J Real Time Image Process*, vol. 20, no. 2, p. 21, 2023, doi: 10.1007/s11554-023-01271-1.

[45] H. A. Santoso, B. F. Safsalta, N. Febrianto, G. W. Saraswati, and S.-C. Haw, "Comparative Analysis of Convolutional Neural Network and DenseNet121 Transfer Learning in Agriculture Focusing on Crop Leaf Disease Identification," *Applied Computing and Informatics*, 2024, doi: 10.1108/aci-03-2024-0132.