UNEC Journal of Computer Science and Digital Technologies
Volume 1, № 1, 2025, pp. 5-17
https://doi.org/10.30546/UNECCSDT.2025.01.005

Section:
Artificial Intelligence &
Machine Learning

Automated Apple Leaf Disease Classification Using Deep Convolutional Neural Networks: A Comparative Study on the Plant Village Dataset

Javanshir Zeynalov[1*], Yiğitcan Çakmak[2], İshak Paçal[3]

[1]Department of Electronics and Information Technologies, Faculty of Architecture and Engineering, Nakhchivan State University, AZ 7012, Nakhchivan, Azerbaijan*
[2, 3] Department of Computer Engineering, Faculty of Engineering, Igdir University, 76000, Igdir, Türkiye
[1]*0009-0002-4985-6371, cavansirzeynalov@ndu.edu.az*
[2]*0009-0008-7227-9182, ygtcncakmak@gmail.com*
[3]*0000-0001-6670-2169, ishakpacal@igdir.edu.tr*

Abstract

The early and accurate identification of plant diseases play a vital role in ensuring agricultural productivity and food security. In this study, we investigate the effectiveness of state-of-the-art convolutional neural network (CNN) architectures for the automated classification of apple leaf diseases using the Plant Village Apple dataset. Five high-performance models DenseNet-264, EfficientNet-B4, EfficientNet-B5, Inception-V3, and MobileNet-V3-Large were fine-tuned on expertly labeled images. DenseNet-264 outperformed other models, achieving an accuracy of 98.32%, precision of 97.83%, recall of 98.21%, and an F1-score of 98.02%. Inception-V3 also demonstrated competitive results, while MobileNet-V3-Large offered a compelling balance between accuracy and computational efficiency, making it suitable for deployment on mobile and edge devices. The findings highlight the capability of deep learning to deliver fast, reliable, and objective diagnostics from ordinary field images, significantly reducing the need for manual inspection. This approach holds promises for enhancing disease management, safeguarding crop yield, and supporting precision agriculture.

***Keywords*:** apple disease detection, deep learning, plant village

Received:	*Revised:*	*Accepted:*	*Published:*
25/05/2025	*01/06/2025*	*06/06/2025*	*14/06/2025*

1. Introduction

Apples are among the most widely grown fruit crops globally, with a total estimated annual production of about 86 million tons in 2020, reflecting their importance in agriculture. In addition to the sheer amount produced, apples are also important for their nutrient density, containing significant amounts of dietary fiber, vitamins, antioxidants, and other health-promoting factors. Even as apples are agriculturally and nutritionally important, apple production faces challenges from foliar diseases that increasingly impact apple crop yields. Major leaf diseases affecting apples include apple scabs, black rot, and cedar apple rust, all of which can have negative impacts on tree health and fruit. For the most part, traditional

diagnoses are made using visual inspection, which is very labor intensive, subjective and allows for diagnostic error depending on observer's experience. As global demand for crops continues to increase, along with the need for sustainable farming methods, early and accurate detection of leaf disease has become increasingly important [1].

In light of these difficulties, artificial intelligence (AI), specifically deep learning (DL), has proven to be a game changer in the diagnostic of plant diseases. Advanced DL architectures, such as Deep Neural Networks (DNNs) or Convolutional Neural Networks (CNNs), are quite adept at visual data analysis and can recognize complex structures and morphology features to ultimately classify plant diseases from images of leaves. Detection methods, such as the polymerase chain reaction (PCR) method, have high specificity, and reliability; however, a PCR typically requires expensive laboratory equipment, trained professionals and, therefore, are often impractical for field-based implementations, given issues with lodging and logistics. DL-based diagnostics and imaging techniques, on the other hand, require only images, allowing for rapid, scalable, and cost-effective detection of disease diagnostics. Traditional machine learning (ML) methods use their own defined features to derive meaning from the data provided and can be used with relatively small datasets, relying on hand-crafted features downstream. In environments that are exceptionally complex, or have high variability, traditional ML methods may not provide the discriminative performance ideal for robust classification. DL takes the approach of learning new hierarchical representations from a raw data response, which allows the model to achieve generalization and flexibility of use. This is particularly valuable when working in low resource conditions for agriculture [2].

The current study applies deep learning models that have been trained with transfer learning, a technique that allows domain-specific tasks with limited data to use models that have previously been trained on large-scale datasets. When training deep learning models use transfer learning, this type of training significantly decreases computation time, and enhances model performance & accuracy, particularly in industry areas like agriculture, where labeled datasets of high quality can be difficult to acquire. By using CNN architecture that has been trained on diverse sets of images such as ImageNet, models can successfully learn complex features to classify apple leaf disease effectively. This leads to more robust and efficient models, which aids in advancing the development of applicable tools powered by AI for precision agriculture.

To establish the relevance of this approach, we now turn to a review of prior research conducted in this domain. Pacal et al. conducted a systematic review of 160 studies published between 2020 and 2024 on deep learning-based plant disease detection, focusing on classification, detection, and segmentation tasks. Their review emphasizes the significant advantages of deep learning approaches over traditional methods, particularly in early and accurate disease identification, thereby offering valuable insights for sustainable agriculture [3]. P et al. introduced the WRLSB-HPS algorithm for plant leaf disease detection using the Plant Village dataset, combining various machine learning techniques such as Logistic Regression, SVM, Naive Bayes, and Random Forest in a weighted ensemble approach to enhance detection accuracy. This method achieved impressive performance, with 98.4% accuracy, 98.2% precision, 97.9% recall, and 97.5% F1-score [4] . Abelonian et al. proposed a hybrid framework combining Convolutional Neural Networks (CNNs) and Vision Transformers (ViT) to enhance classification accuracy. The ensemble model, which incorporates VGG16, Inception-V3, and DenseNet20 architectures for global feature extraction and ViT for local feature capture, demonstrated superior performance on two publicly available datasets (Apple and Corn),

achieving accuracy rates of 99.24% and 98%, respectively [5] Gupta et al. introduced Plant DetectNet, a hybrid deep learning model intel rated with the Internet of Things (IoT) to improve plant disease detection. The framework k utilized sensor data and images from the Plant Village dataset, employing techniques like GRU for temporal feature extraction, Depthcat CNN for spatial features, and a Gated ConvNeXt model for enhanced classification. This approach achieved 98.8% accuracy and 95.9% recall, outperforming existing methods and demonstrating scalability and efficiency in plant disease detection [6] .

The review of the literature indicates that deep learning methods have significant benefits for plant disease detection, and with improvements over traditional methods. When trained on big datasets, these methods can accurately identify diseases in plants. The largest advantages included early detection, accuracy, cost, and speed. The applications of transfer learning allow models to still perform well with smaller datasets. The combination of deep learning with new technologies such as IoT and large datasets also increases their capabilities. Ultimately, the role of deep learning models in detection and management of plant diseases will become increasingly crucial and provide more advancements for agricultural practices in the future.

2. Material and Methods

2.1 Plant Village dataset

The quality and structure of the dataset are crucial determinants of the effectiveness of deep learning models. In contrast to traditional machine learning methods, which typically rely on manual feature extraction and smaller datasets, deep learning models necessitate large, high-quality datasets to effectively capture meaningful and distinguishing features from raw data. This is essential to ensure that deep models generalize effectively and deliver strong predictive performance. Table 1. outlines various components of this study's dataset, which was partitioned into training, validation, and test sets to facilitate thorough model evaluation and prevent data leakage during the learning process [7] .

	Images	%
Train	2219	70
Test	477	15
Validation	475	15
Total	3171	100

Table 1. Distinction between train test and validation

The Plant Village Apple dataset was selected as the main data source for the research, as it is an established, publicly available benchmark dataset for plant disease identification. The Plant Village dataset is known for its wide coverage and high resolution and contains data for a variety of plant species and disease classes, including healthy and diseased. This research focused only on the apple subset for classification of disease. The total sample size of the dataset being used was 3,171, which were partitioned into training, validation, and testing sets for evaluating the model (70% training, 15% testing, 15% validation) [8].

Figure 1. presents a curated selection of image samples from the PlantVillage dataset utilized in this study, illustrating the visual characteristics of different apple leaf conditions.

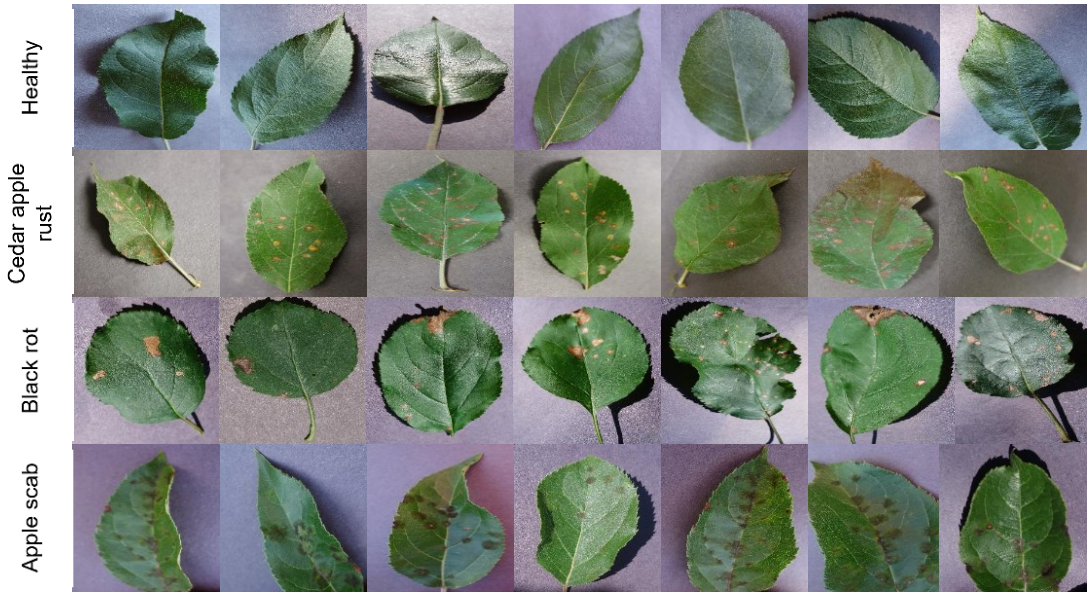

Figure 1. Examples from our dataset

The figure is organized into four distinct categories, displayed row-wise: "Healthy" leaves, exhibiting no visible signs of disease; leaves afflicted with "Cedar apple rust," typically characterized by orange or rust-colored lesions; leaves showing symptoms of "Black rot," often presenting as dark, necrotic spots or lesions; and leaves infected with "Apple scab," which commonly manifests as dark, olive-green to black, velvety spots. Each category shows multiple examples, highlighting the intra-class visual variability and the specific pathological symptoms that the deep learning models were trained to differentiate. This visual representation underscores the diversity of the dataset and the challenges inherent in automated disease classification.

2.2 Data Augmentation

Data augmentation is an accepted process in deep learning to enhance model performance and is particularly useful for situations where you do not have access to an extremely large dataset and a wide variety of data. The data augmentation process artificially increases the training dataset by performing a few transformations which will have the effect of modifying or control the variation in the data. Data augmentation can help reduce overfitting, particularly when the dataset is small or unbalanced, and it aids in better generalization to unseen data. For this research study, a strong data augmentation pipeline was used during the preprocessing stage to increase the model robustness and prediction accuracy. The data augmentation transformations used in this research study are: RandomResizedCrop which randomly crops and resizes an image to have a fixed size of 224x224, which encourages spatial variation and averts overfitting to a certain part of the image; RandomHorizontalFlip encourages the model to learn features that are invariant to orientation, by randomly applying horizontal flipping; Random Rotation randomly rotates the image in a rotationally invariant manner from ±15 degrees which can simulate captured image
from different orientations and helps the model to become more invariant to rotation; ColorJitter to simulate various changes in lighting scenarios and environments, randomly adjusting the brightness, contrast, saturation and hue values. The images were converted into PyTorch tensors using ToTensor and normalized with ImageNet statistics for consistent and stable training, and

faster convergence.These augmentations are anticipated to significantly reduce the risk of overfitting, improve generalization, and ultimately increase the model's predictive performance.

2.3 Deep Learning Architectures

Machine learning has revolutionized technological progress and human advancement, becoming a key driver in various modern applications, such as enhancing search engine capabilities, moderating user-generated content on social media, and powering personalized recommendation systems for e-commerce. As technology rapidly evolves, machine learning methods are increasingly integrated into everyday life, manifesting in smart technologies and advanced systems with capabilities like visual object detection, speech recognition, and adaptive dynamic content in digital environments [9] .

The rapid advancements in artificial intelligence are largely attributed to the evolution of deep learning. A specialized branch of machine learning, deep learning utilizes intricate, multi-layered neural networks to derive complex, non-linear representations from vast datasets. These models identify detailed features through hierarchical structures and are trained using backpropagation. Deep learning has proven exceptionally successful across multiple fields, including image and video analysis, speech processing, and natural language understanding. Convolutional Neural Networks (CNNs) excel in processing spatial data, while Recurrent Neural Networks (RNNs) are ideal for handling temporal or sequential data such as speech and text [10, 11] .

Although Geoffrey Hinton introduced the fundamental principles of deep learning in 2006, its broad adoption came after deep models drastically outperformed traditional algorithms in the ImageNet Large Scale Visual Recognition Challenge. Since then, deep learning has consistently provided cutting-edge results in a wide range of applications, such as pattern recognition, classification, forecasting, drug discovery, signal analysis, finance, healthcare, and defense, and it remains the leading paradigm in both AI research and practical deployment [12].

Figure 2. delineates a generalized algorithmic workflow for training a Convolutional Neural Network (CNN). The process commences with the initialization of the CNN architecture, followed by an iterative training phase over multiple epoch. Within each epoch, the training data is processed in mini batches. For every mini batch, a forward pass is executed, where input data traverses through the network layers to compute output feature maps, apply activation functions, and perform pooling operations. Subsequently, the feature maps are flattened, and a loss function quantifies the discrepancy between the predicted and actual labels. This loss is then utilized in the backward pass to compute gradients with respect to the model parameters, which are subsequently updated using an optimization algorithm such as SGD or Adam, ultimately yielding a trained CNN model. This structured approach, involving iterative forward and backward propagation, is fundamental to the learning capability of CNNs in visual recognition tasks.

```
Input: Training data, labels, hyperparameters
Output: Trained CNN model
Initialize CNN architecture
for each training epoch do
  for each mini-batch of training samples do
    Forward pass:
    for each layer in the CNN do
      Compute the output feature maps
      Apply activation function
      Apply pooling (if applicable)
    end for
    Flatten the feature maps
    Compute the loss between predicted and actual labels
    Backward pass:
    Compute the gradient of the loss with respect to the parameters
    Update the parameters using an optimizer (e.g., SGD, Adam)
  end for
end for
return Trained CNN model
```

Figure 2. A typical CNN algorithm [13].

2.4 Algorithms Used

EfficientNet B4 leverages a compound scaling technique that simultaneously adjusts the network's depth, width, and input resolution in a principled manner. By uniformly scaling these dimensions, it achieves a high level of classification accuracy without incurring significant computational overhead. EfficientNet B4 has proven to be an effective and scalable architecture, frequently outperforming traditional CNNs on diverse image recognition tasks [14] .

EfficientNet B5 represents an enhanced variant within the EfficientNet model family, incorporating a more extensive scaling of architectural dimensions. With increased layer depth, wider convolutional channels, and larger input sizes, EfficientNet B5 enables richer feature extraction. It maintains computational efficiency while delivering superior accuracy, making it well-suited for complex vision applications that demand high performance [14].

ResNet 50 consists of 50 convolutional layers and introduces the concept of residual learning through shortcut connections, which bypass one or more layers. These connections allow the network to effectively mitigate the vanishing gradient issue, thereby enabling the training of much deeper models. ResNet 50 has become a foundational architecture in computer vision due to its robust performance and ease of optimization on large-scale datasets such as ImageNet [15]. ResNet variants and their layers are shown in Figure 3.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3. Layer-wise architectural specifications for ResNet variants (18-layer to 152-layer), detailing convolutional block configurations, output feature map dimensions, and computational complexity in FLOPs.

DenseNet 264 adopts a densely connected design where each layer receives the concatenated outputs of all preceding layers. This architectural strategy promotes efficient feature propagation and reusability, which leads to improved model compactness and faster convergence. DenseNet 264 has exhibited exceptional accuracy in fine-grained image classification tasks and is recognized for its parameter efficiency despite its depth [16]. DenseNet variants and their layers are shown in Figure 4.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix}\times48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 4. Architectural details of DenseNet variants (DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264), outlining layer types, output sizes, dense block configurations, and transition layers.

Inception v3 incorporates a modular design that utilizes techniques such as convolution factorization, auxiliary classifiers for regularization, and reduced dimensionality in intermediate layers. These innovations collectively enhance computational efficiency and model expressiveness. Inception v3 remains a widely adopted solution for visual recognition problems, particularly those requiring multi-scale feature abstraction and efficient inference [17].

MobileNet v3-large is an advanced lightweight CNN architecture tailored for low-power and real-time applications. It combines depthwise separable convolutions with squeeze-and-excitation modules and adopts the hard-swish activation to improve performance without increasing computational cost. This design enables MobileNet v3-large to deliver strong accuracy while remaining suitable for deployment on mobile and embedded platforms with limited resources [18].

3. Results and Discussions

3.1 Experimental Design

The experiments presented in this study were conducted on a Linux system with Ubuntu 22.04, featuring an Intel Core i5-13600K CPU, 32 GB DDR5 RAM, and an NVIDIA RTX 3090 GPU. All models were developed using PyTorch with NVIDIA's CUDA augmentation. The models were trained and evaluated under consistent experimental conditions and with the same set of hyperparameters to ensure standardization and systematic comparison.

3.2 Performance Metrics

Evaluating the performance of deep learning models is an essential step for assessing their usefulness, justifying related decisions, and facilitating data-driven choices. Performance metrics can serve several primary purposes, like assessing the performance of classification models, assisting optimization, identifying errors or biases in reports of the data, comparing models, and detecting overfitting. We have specifically focused on performance metrics for grape disease classification in this paper and made conventional choices for evaluation criteria that are rigorously established in academic literature.

The fundamental metrics used in this project accuracy, precision, recall, and F1-score are key components in deep learning evaluation as well in other areas. Accuracy is defined as the proportion of correctly classified samples compared to the total number of samples and provides a view of overall performance. Precision (true positives/(true positive + false positives)) indicates how reliable the model is in classifying relevant instances (high precision = few or no false positives) and recall shows which instances the model identified as true positives, and recall is relevant for measuring relevance. The F1-score is defined as the harmonic means of precision and recall, combining them as a single measure of performance (also relevant for false positives and false negatives). However, while these definitions can feel complicated, they can be represented mathematically as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Number\ of\ total\ predictions}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3.3 Results

In this study, we evaluated the performance of various advanced convolutional neural network (CNN) architectures, including DenseNet-264, EfficientNet-B4, EfficientNet-B5,

Inception-V3, MobileNet-V3-Large, and ResNet-50, specifically applied to apple leaf disease classification using the Plant Village dataset. Each model was fine-tuned under identical experimental conditions, including consistent hyperparameters, to ensure reliable and systematic comparison. The evaluation employed several key metrics widely recognized in the literature, namely accuracy, precision, recall, and F1-score, each providing complementary insights into different aspects of the models' predictive capabilities. These metrics are essential in assessing the overall effectiveness, reliability, and applicability of CNN models in practical agricultural scenarios.

Table 2 summarizes the comparative performance of the CNN models. DenseNet-264 exhibited the highest performance across all evaluated metrics, achieving an impressive accuracy of 98.32%, precision of 97.83%, recall of 98.21%, and an F1-score of 98.02%. This outcome highlights DenseNet-264's superior capability in accurately distinguishing between different apple leaf disease categories, significantly outperforming other architectures evaluated in this study.

Models	Accuracy %	Precision %	Recall %	F1-Score %
DenseNet-264	98.32	97.83	98.21	98.02
EfficientNet-B4	93.08	90.74	91.04	90.62
EfficientNet-B5	93.71	91.96	92.12	92.01
İnception-V3	97.90	97.73	97.65	97.66
MobileNet-V3-Large	93.71	94.73	93.17	93.91
ResNet-50	93.71	92.05	92.54	92.24

Table 2. Results of CNN-based models on PlantVillage dataset

Inception-V3 also delivered highly competitive results, attaining an accuracy of 97.90% along with robust precision, recall, and F1-score metrics. Its ability to handle complex feature extraction tasks efficiently is underscored by the minimal misclassification observed in the confusion matrix. Inception-V3's performance closely rivals that of DenseNet-264, confirming its effectiveness in detailed and accurate image analysis.

EfficientNet variants demonstrated commendable but slightly lower performance. Specifically, EfficientNet-B4 achieved 93.08% accuracy with precision and recall of approximately 91%. EfficientNet-B5 improved slightly upon this performance, registering an accuracy of 93.71%, precision of 91.96%, and recall of 92.12%. While both EfficientNet models showed good training stability and satisfactory classification results, their performance was not as robust as DenseNet-264 or Inception-V3, particularly in handling the nuanced features of apple leaf diseases.

MobileNet-V3-Large stood out by effectively balancing computational efficiency and predictive performance. With an accuracy of 93.71% and precision of 94.73%, MobileNet-V3-Large proved suitable for deployment in resource-constrained environments such as mobile and edge devices, making it a practical choice for field applications. Although it exhibited slightly higher confusion rates than DenseNet-264 and Inception-V3, its lightweight nature provides significant advantages for practical implementations.

ResNet-50, despite its popularity and widespread use in other applications, provided moderate performance in this context, achieving an accuracy of 93.71%. While its precision and recall metrics were reasonably balanced, the overall effectiveness in distinguishing between nuanced

apple leaf conditions was not as pronounced, indicating limitations potentially related to its architectural design or depth.

Demonstrating its superior capabilities, DenseNet-264, along with Inception-V3, showed rapid and stable convergence during training and achieved minimal confusion in classifying the apple leaf images, with DenseNet-264 particularly excelling in these aspects. EfficientNet variants exhibited consistent but slightly slower improvement trajectories, while MobileNet-V3-Large demonstrated efficiency with acceptable levels of confusion. ResNet-50 had steady performance throughout training but encountered more difficulties in accurate disease classification. The training graph of our DenseNet-264 model, which demonstrated the most superior performance, is shown in Figure 6, and its confusion matrix is shown in Figure 7.

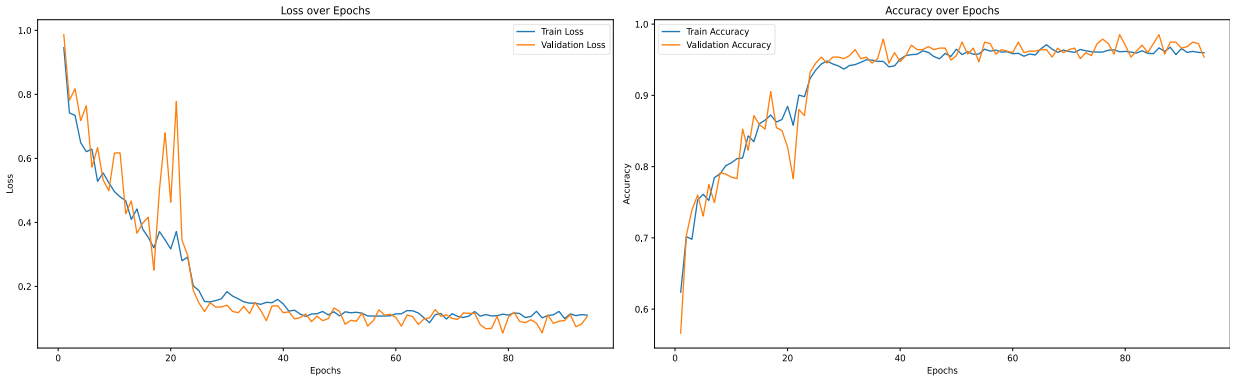

Figure 6. Training and validation loss (left) and accuracy (right) curves over epochs for the DenseNet-264 model, illustrating learning progression.

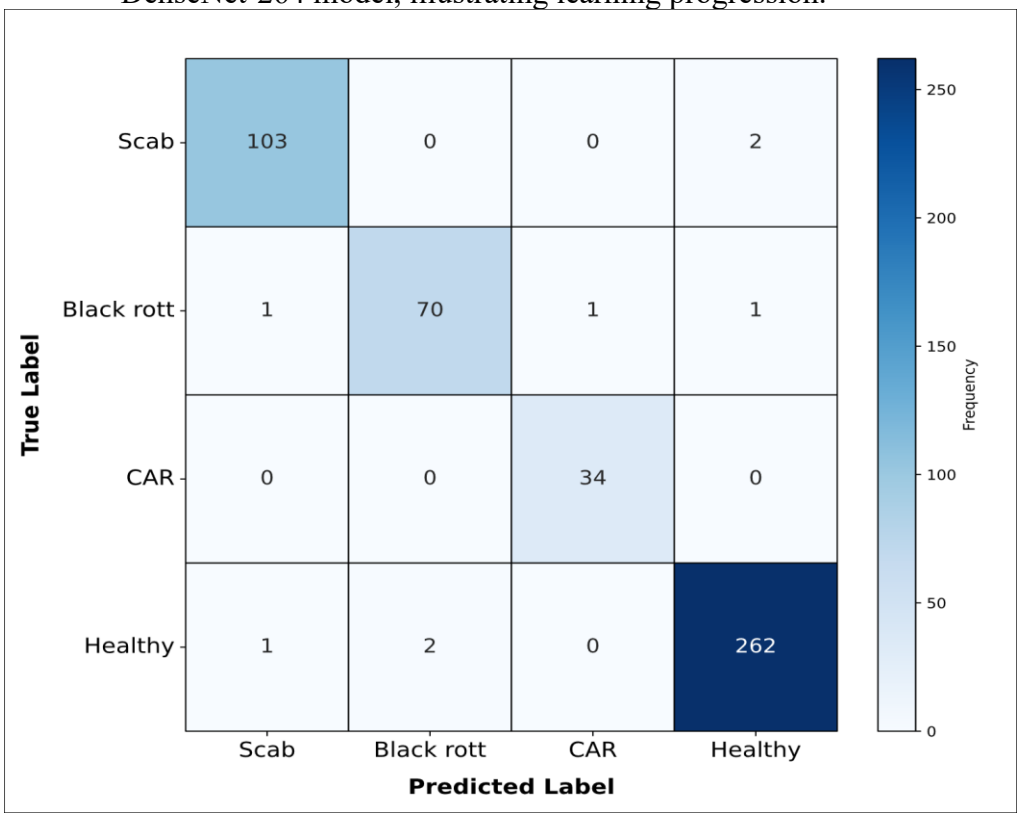

Figure 7. Confusion matrix for the DenseNet-264 model, illustrating classification performance across apple leaf disease categories (Scab, Black rot, Cedar Apple Rust (CAR), and Healthy).

3.4 Discussion

The results of this study underscore the considerable potential of deep learning-based models for automating apple leaf disease detection, thereby supporting precision agriculture. DenseNet-264, with its densely connected structure that enhances gradient flow and feature reuse, stood out as the most effective model, delivering both high accuracy and balanced classification performance across metrics. These results highlight their aptitude for handling complex image patterns present in disease-affected leaves.

Inception-V3 also demonstrated robust performance, benefiting from its multi-scale feature abstraction capability. Its strong results confirm the model's ability to generalize well across the dataset, making it a viable candidate for scenarios requiring high reliability. The high precision and recall achieved suggest that both DenseNet-264 and Inception-V3 could substantially reduce false diagnoses, thereby improving disease management efficiency.

MobileNet-V3-Large's performance is especially relevant for real-world applications, where computing resources may be limited. Its lightweight architecture and acceptable accuracy levels indicate that it can serve as an effective model for on-device diagnosis tools, such as mobile applications used by farmers in the field. This extends the impact of AI tools from research settings to practical, everyday agricultural use.

Although EfficientNet-B4 and B5 performed moderately well, their results suggest that their compound scaling strategy may not capture the complex visual characteristics of apple leaf diseases as effectively as DenseNet or Inception. Meanwhile, ResNet-50, despite its historical success, faced challenges in distinguishing subtle variations among disease categories, possibly due to insufficient depth or lack of advanced feature extraction modules.

The overall findings validate the utility of CNNs in plant pathology and reinforce the need for model selection based on the deployment context. While accuracy is critical, factors such as model size, inference speed, and ease of integration into portable systems should guide the choice of architecture for operational use. Future work should consider the generalizability of these models under variable field conditions, including different lighting, background, and occlusion scenarios. Further, the integration of explainable AI techniques could enhance model interpretability, increase user trust and facilitate the adoption of AI-based solutions in agriculture. Finally, testing these models on other apple cultivars and regional datasets would provide additional validation and support global scalability.

4. Conclusion

This research explores the performance of advanced convolutional neural networks in the automated detection of apple leaf diseases using the PlantVillage Apple dataset. Five cutting-edge models DenseNet-264, EfficientNet-B4, EfficientNet-B5, Inception-V3, and MobileNet-V3-Large were fine-tuned on expertly annotated apple leaf images. Among them, DenseNet-264 delivered the most impressive results, achieving 98.32% accuracy, 97.83% precision, 98.21% recall, and a 98.02% F1-score. Inception-V3 also showed strong performance, closely rivaling the top model, while MobileNet-V3-Large proved to be a practical choice for on-device applications due to its efficient architecture and competitive accuracy. These results emphasize the strength of deep learning in converting everyday agricultural images into rapid, accurate, and consistent diagnostic tools. By minimizing reliance on manual inspections, the models offer

a reliable approach to timely and economic disease detection, ultimately supporting better crop protection and yield improvement.

Acknowledgment
We would like to express our gratitude to the staff of the Department of Digital Technologies and Applied Informatics of the Azerbaijan State University of Economics for their assistance in researching materials on the problem.

Authors' Declaration
Conflicts of Interest: The authors declare no conflict of interest.

Authors' Contribution Statement
All authors contributed equally to this work.

References

1. D. Rohith, P. Saurabh, D. Bisen, An integrated approach to apple leaf disease detection: leveraging convolutional neural networks for accurate diagnosis, Multimed Tools Appl (2025) 1–36. https://doi.org/10.1007/S11042-025-20735-Z/TABLES/11.
2. A. Banjar, A. Javed, M. Nawaz, H. Dawood, E-AppleNet: An Enhanced Deep Learning Approach for Apple Fruit Leaf Disease Classification, Applied Fruit Science 67 (2025) 1–11. https://doi.org/10.1007/S10341-024-01239-W/TABLES/4.
3. I. Pacal, I. Kunduracioglu, M.H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, R. Martinek, A systematic review of deep learning techniques for plant diseases, Artificial Intelligence Review 2024 57:11 57 (2024) 1–39. https://doi.org/10.1007/S10462-024-10944-7.
4. K. P, H. Lalitha, D.J. Priya, B.S. C, AI-driven plant health monitoring: evaluating the WRLSB-HPS algorithm for leaf disease classification, Earth Sci Inform 18 (2025) 1–18. https://doi.org/10.1007/S12145-025-01762-8/TABLES/6.
5. S. Aboelenin, F.A. Elbasheer, M.M. Eltoukhy, W.M. El-Hady, K.M. Hosny, A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer, Complex and Intelligent Systems 11 (2025) 1–17. https://doi.org/10.1007/S40747-024-01764-X/TABLES/6.
6. P. Gupta, · Rakesh, S. Jadon, PLANT Detect Net: a hybrid IoT and deep learning framework for secure plant disease detection and classification, Evolving Systems 2025 16:2 16 (2025) 1–20. https://doi.org/10.1007/S12530-025-09685-X.
7. I. Pacal, MaxCerVixT: A novel lightweight vision transformer-based Approach for precise cervical cancer detection, Knowl Based Syst 289 (2024) 111482. https://doi.org/10.1016/J.KNOSYS.2024.111482.
8. David.P. Hughes, M. Salathe, An open access repository of images on plant health to enable the development of mobile disease diagnostics, (2015). https://arxiv.org/abs/1511.08060v2 (accessed April 20, 2025).
9. Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 2015 521:7553 521 (2015) 436–444. https://doi.org/10.1038/nature14539.
10. A. Karaman, I. Pacal, A. Basturk, B. Akay, U. Nalbantoglu, S. Coskun, O. Sahin, D. Karaboga, Robust real-time polyp detection system design based on YOLO algorithms by

optimizing activation functions and hyper-parameters with artificial bee colony (ABC), Expert Syst Appl 221 (2023) 119741. https://doi.org/10.1016/J.ESWA.2023.119741.

11. I. Pacal, A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu, S. Coskun, An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets, Comput Biol Med 141 (2022) 105031. https://doi.org/10.1016/J.COMPBIOMED.2021.105031.
12. I. Pacal, enhancing crop productivity and sustainability through disease identification in maize leaves: Exploiting a large dataset with an advanced vision transformer model, Expert Syst Appl 238 (2024) 122099. https://doi.org/10.1016/J.ESWA.2023.122099.
13. I. Kunduracioglu, I. Pacal, Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases, Journal of Plant Diseases and Protection 131 (2024) 1061–1080. https://doi.org/10.1007/S41348-024-00896-Z/TABLES/7.
14. M. Tan, Q. V Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, (2020).
15. K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, (2015). http://arxiv.org/abs/1512.03385.
16. G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, (2016). http://arxiv.org/abs/1608.06993.
17. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Rethinking the Inception Architecture for Computer Vision, (2015).
18. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V Le, H. Adam, G. Ai, G. Brain, Searching for MobileNetV3, (2019).