



OPEN

A fine tuned EfficientNet-B0 convolutional neural network for accurate and efficient classification of apple leaf diseases

Hassan Ali^{1,2}, Noora Shifa², Rachid Benlamri¹, Aitazaz A. Farooque^{3,5} & Raziq Yaqub⁴

Precise classification and detection of apple diseases are essential for efficient crop management and maximizing yield. This paper presents a fine-tuned EfficientNet-B0 convolutional neural network (CNN) for the automated classification of apple leaf diseases. The model builds upon a pre-trained EfficientNet-B0 base, enhanced through architectural modifications such as the integration of a global max pooling (GMP) layer, dropout, regularization, and full-model fine-tuning. To address class imbalance and improve generalization, the study adopts a holistic training strategy that integrates data augmentation, stratified data splitting, and class weighting, alongside transfer learning. The model is evaluated on the PlantVillage (PV) dataset and a curated Apple PV (APV) dataset and compared against EfficientNet-B0, EfficientNet-B3, Inception-v3, ResNet50, and VGG16 models. The fine-tuned model demonstrates outstanding test accuracies of 99.69% and 99.78% for classifying plant diseases using the APV and PV datasets, respectively. The fine-tuned model outperforms EfficientNet-B0, EfficientNet-B3, and VGG16 on both datasets and shows superior performance compared to Inception-v3 and ResNet-50 on the PV dataset. Both EfficientNet-B0 and the fine-tuned model demonstrate the lowest memory consumption and floating-point operations per second (FLOPs). Also, as compared to the EfficientNet-B0 model, the fine-tuned model achieves an 11% increase in accuracy on the APV dataset and a 49.5% accuracy improvement on the PV dataset, with approximately a 7-8% increase in both memory usage and FLOPs. The fine-tuned model thus emerges as an effective solution for plant leaf disease classification, delivering outstanding accuracy with optimized memory consumption and FLOPs, making it suitable for resource-constrained environments. This study demonstrates that fine-tuned CNN approaches, when combined with transfer learning, advanced data pre-processing, and architectural optimizations, can significantly enhance the accuracy of diseased leaf classification in crops with efficient implementation in limited-resource settings.

Keywords Apple leaf diseases, EfficientNet-B0, Deep learning, CNN, Transfer learning, Fine-tuning

The worldwide cultivation of fruits and crops is faced with significant challenges as a result of various diseases, which subsequently have an economic impact on the agricultural industry. Apples are widely grown fruits with a significant impact on the global market. In 2022, the worldwide output of apples reached over 95 million metric tons, with China contributing almost half of this amount¹. However, despite the strong demand for apples, the actual amount produced often does not reach its full potential due to many problems, such as ecological circumstances, insufficient post-harvest technologies, limited research emphasis, and socio-economic limitations². Apple trees are highly vulnerable to a range of diseases caused by insects and microorganisms. Several prevalent diseases affecting apple trees include cedar-apple rust, black rot, fire blight, apple scab, root rot, and powdery mildew^{3,4}. Failure to control these diseases can have a substantial effect on the quantity of fruit

¹Centre of Excellence for Sustainability and Food Security, University of Doha for Science and Technology, Doha 24449, Qatar. ²Department of Electrical Engineering, University of Doha for Science and Technology, Doha 24449, Qatar. ³Canadian Centre for Climate Change and Adaptation, University of Prince Edward Island, St Peters Bay, PE, Canada. ⁴Department of Electrical Engineering and Computer Science, Alabama A&M University, Huntsville, AL 35762, USA. ⁵Faculty of Sustainable Design Engineering, University of Prince Edward Island, Charlottetown, PE, Canada. email: hassan.ali@udst.edu.qa

produced. Therefore, accurate diagnosis and suitable treatments are crucial strategies to manage these diseases and consequently reduce agricultural losses.

Conventional diagnosis techniques for plant leaf diseases, which mostly rely on visual examinations, are frequently expensive and ineffective, resulting in missed chances for disease prevention. The lack of automated systems for disease detection and classification worsens these issues, requiring regular expert monitoring and consultations, leading to inefficient use of resources and decreased fruit quality. Advancements in technology, namely in machine learning (ML)^{5–9} and deep learning (DL)^{2,10–23}, provide potential options for automating the detection and classification of plant diseases. DL techniques, particularly convolutional neural networks (CNNs), have made notable advancements in automating the classification and detection of diseases in crops²⁴. CNNs can efficiently extract important features from images without the need for explicit feature extraction or complex image processing steps. As a result, they can make highly accurate predictions²⁵.

Technology is crucial in the field of DL-based plant disease classification and detection. Recently, substantial developments in graphics processing units (GPUs) have revolutionized the efficiency and speed of DL training. Nevertheless, due to the growing demand for real-time classification and detection while maintaining high accuracy and efficiency, there is a rising need for more efficient and computationally lightweight DL techniques. Accordingly, when developing a CNN model, developers and researchers usually concentrate on three main areas:

- **Memory reduction:** To deploy the model in environments with limited hardware resources, it is essential to reduce the amount of memory consumed by the model.
- **Minimizing floating-point operations:** Minimizing the number of floating-point operations per second (FLOPs) is crucial for improving the inference speed of the model.
- **Improved accuracy:** The model must have highly accurate detection and classification performance to be practically useful.

Recent advancements in DL have revolutionized plant disease classification and detection, yet deploying these models in resource-constrained agricultural settings remains a significant challenge. Existing DL models often demand substantial memory and computational resources, limiting their practicality in such environments. To address these limitations, this study aims to develop a CNN-based model for apple leaf disease classification that achieves high accuracy while being memory and FLOPs efficient. By prioritizing these factors, our goal is to create a model suitable for situations with limited resources and practical applications, without compromising cutting-edge performance.

EfficientNets, developed by the Google Research Team, Tan and Le, in May 2019²⁶, are a notable achievement in terms of accuracy and efficiency, accomplished through a process of neural architecture search and subsequent scaling^{26,27}. The EfficientNet series consists of eight variants, ranging from B0 to B7, which have been scaled using a compound scaling approach. This method utilizes a coefficient ϕ to uniformly alter the depth, width, and resolution of the models^{26,28}. Notably, the baseline EfficientNet-B0 variant strikes a balanced trade-off between model complexity and performance, making it well-suited for a diverse array of computer vision tasks, particularly those with restricted computational resources. Also, its efficient architecture enables faster training and inference times while maintaining good accuracy.

This paper introduces a fine-tuned CNN model that utilizes the EfficientNet-B0 basic model architecture, designed specifically for the apple leaf disease classification. To address the issue of class imbalance and improve generalization, the study employs a holistic training strategy integrating data augmentation^{29,30}, stratified data splitting^{31,32}, and class weighting³³, alongside transfer learning. To enhance the model performance without increasing memory and FLOPs, a pre-trained EfficientNet-B0 is utilized. To further enhance the classification accuracy, the higher layers of the base model are modified and subsequently the entire model architecture is fine-tuned for classification leaf diseases. A comprehensive set of comparative experiments are carried out to evaluate the fine-tuned model's performance on both the PlantVillage (PV) dataset³⁴ and the Apple PV (APV) dataset. The APV dataset is a carefully selected collection of apple leaf diseases (cedar-apple rust, apple scab, black rot) as well as healthy leaves from the PV dataset. The performance of the proposed model is also compared it with five existing state-of-the-art pre-trained CNN classification models: EfficientNet-B0²⁶, EfficientNet-B3²⁶, Residual Network-50 (ResNet-50)³⁵, Visual Geometry Group 16 (VGG16)^{36–39}, and Inception-v3⁴⁰.

The fine-tuned EfficientNet-B0 model demonstrated outstanding precision, recall, and F1 scores, obtaining test accuracies of 99.69% and 99.78% for identifying plant diseases using the APV and PV datasets, respectively. The fine-tuned model achieved superior performance compared to EfficientNet-B0, EfficientNet-B3, and VGG16 on both the datasets. Additionally, it outperformed Inception-v3 and ResNet-50 on the PV dataset. Both EfficientNet-B0 and fine-tuned EfficientNet-B0 stood out as the most efficient solutions in terms of memory usage and FLOPs requirements compared to the other models. Nevertheless, the fine-tuned EfficientNet-B0 model demonstrated its superiority over the EfficientNet-B0 model by achieving an 11% increase in accuracy on the APV dataset and an amazing 49.5% improvement on the PV dataset. Remarkably, the fine-tuned EfficientNet-B0 model's improved performance resulted in an approximate 7–8% rise in both memory consumption and FLOPs (as compared with the EfficientNet-B0 model requirements), making it an excellent choice for precise plant leaf disease classification while maintaining a balance between performance and resource efficiency.

Novelty and contributions: Our work offers several novel contributions:

- **Architectural optimization with global max pooling (GMP):** A novel architectural modification was implemented by replacing the global average pooling (GAP) layer with GMP in the EfficientNet-B0 model. This adjustment prioritizes disease-specific, localized patterns such as lesions and spots, enabling finer feature discrimination.

- **High efficiency in resource-constrained environments:** The model achieves exceptional performance with minimal memory and FLOPs usage, making it suitable for real-world deployment in resource-limited settings.
- **Fine-tuning strategy:** Through transfer learning and fine-tuning of the EfficientNet-B0 architecture, our model significantly outperforms state-of-the-art models, including EfficientNet-B3, Inception-V3, ResNet-50, and VGG16.
- **Generalizability:** By addressing data imbalance with techniques like data augmentation, stratified data splitting, class weighting, and transfer learning, our model demonstrates robustness across varied datasets and holds potential for broader agricultural applications.

Related works

ML methods

A range of ML algorithms have been proposed to detect and classify apple leaf diseases, yielding promising outcomes.

For example, a segmentation technique for apple leaf spot disease, serving as a pre-processing step in ML-based leaf disease classification, was introduced based on the particle swarm optimization (PSO) and K-means PSO (PSOK) algorithm⁶. The PSOK approach demonstrated superior performance in effectively segmenting apple leaf spot disease by converting the RGB leaf image to the CIE L*a*b format, performing clustering on the 'a' component using PSO, and employing the global best of PSO as the initial centroid for K-means, surpassing the conventional K-means algorithm. However, PSO and PSOK techniques involve computationally expensive iterative processes. This limits scalability and makes these techniques unsuitable for real-time or resource-constrained environments.

An improved computerized approach for accurately segmenting and classifying apple leaf diseases was presented in⁷. This approach relied on correlation and genetic algorithm-based feature selection, emphasizing the significance of feature extraction in disease identification. The authors enhanced the visibility of apple leaf spots by applying various image processing techniques, including a 3D box filter, de-correlation, a 3D-Gaussian filter, and a 3D-median filter. The process of segmenting lesion locations was accomplished through the use of a robust correlation-based technique. The process was enhanced by incorporating expectation maximization (EM) segmentation. Subsequently, the color, color histogram, and local binary pattern (LBP) information were merged using comparison-based parallel fusion. The collected features were optimized through a genetic technique and then classified using a one-vs-all multi-class support vector machine (SVM)⁴¹. Although these approaches demonstrated impressive results, their scalability for real-time applications is limited by their heavy reliance on genetic optimization and multi-stage feature extraction methods, which raise processing needs considerably.

A method to classify various leaf diseases by utilizing a set of 8750 photos, encompassing 23 different classes, from the PV dataset, was devised in⁴². The authors employed a pre-trained AlexNet CNN⁴³ to extract features and subsequently employed PSO algorithm to choose the 34 most informative characteristics from the retrieved dataset. Multiple classifiers were then used for the classification task. However, the dependency on the AlexNet and PSO, both of which are computationally intensive, limits the practicality of this approach for resource-constrained environments.

A comprehensive study was conducted on 400 apple leaf images in⁸. The color and texture characteristics were retrieved from the images and the predictors were narrowed down to R, V, and b*. The researchers performed a comparative analysis of K-nearest neighbor (KNN), artificial neural network (ANN), and Gaussian process regression (GPR) models to ascertain healthy and infected apple leaves. The results demonstrated that the GPR model employing the automatic relevance detection (ARD) squared kernel function exhibited superior performance in detecting leaf health. Also, a comparison was made between decision tree and SVM to determine the superior model for disease classification. As compared to the decision tree approach, the quadratic SVM model demonstrated superior classification accuracy in identifying apple leaf diseases. However, handcrafted feature extraction techniques often fail short of capturing the full complexity of leaf diseases, especially for unseen datasets or diverse conditions. This limitation reduces the model's robustness and adaptability in practical scenarios.

In⁹, Chakraborty *et. al* employed a range of image processing techniques, including the Otsu thresholding algorithm and histogram equalization, for image pre-processing. Image segmentation was then employed to extract the diseased regions. Subsequently, a multi-class SVM was used to classify the disease type from a dataset of 500 leaf images. This approach achieved a remarkable accuracy of 96%. Despite its high accuracy, the computational overhead of pre-processing steps and the limited scalability of SVM models present challenges for real-world implementation.

Similarly, Sai and Patel⁵ utilized Hu moments, Haralick texture, and color histogram techniques for feature extraction to classify diseased apple leaves. A comparison was performed on seven ML models, including logistic regression, linear discriminant analysis, KNN, decision trees, random forest, Naive Bayes, and SVM models. The random forest model outperformed the other ML models with a test accuracy of 98.125%. While this study highlights the effectiveness of random forest for disease classification, the reliance on handcrafted feature extraction limits adaptability and scalability, especially for unseen data.

DL methods

ML techniques often require explicit feature extraction or complex image processing steps, which can limit their accuracy and scalability. Advances in DL have revolutionized the field by automating feature extraction and achieving higher detection and classification performance.

In recent years, DL techniques, particularly CNNs, have been increasingly utilized for detection and classification of apple leaf diseases. Detection methods, particularly those leveraging advanced DL models, have significantly evolved, enabling the precise identification and localization of diseased regions on apple leaves with high precision.

For example, a noteworthy approach was introduced in⁴⁴, where an integrated neural architecture for real-time single shot multi-box detector (INAR-SSD) was proposed for real-time diagnosis, achieving a mean average precision (mAP) of 78.80% with a processing speed of 23.13 frames per second (FPS).

An improved YOLOv5-based model, A-Net, has been proposed in²¹ for apple leaf disease detection. The model integrates the Wise-IoU loss function, attention mechanisms, and the RepVGG module, achieving a mAP of 92.7%.

Similarly, in²², a DF-Tiny-YOLO DL model was introduced to improve both the speed and effectiveness of automated apple leaf disease detection. The DF-Tiny-YOLO model achieved a mAP of 99.99% and an average intersection over union (IoU) of 90.88%. Also, the model demonstrated a detection rate of 280 FPS. The improved model demonstrated a significant improvement in detecting apple leaf diseases compared to the Tiny-YOLO and YOLOv2 network models.

DL-based leaf disease detection models provide comprehensive information by combining localization and classification, making them valuable for precision agriculture tasks. These models output bounding boxes and class labels, enabling targeted interventions like spot spraying. However, their high computational complexity and resource requirements can hinder their real-time applications, particularly in resource-constrained settings.

In contrast, classification models focus on categorizing disease types based on symptoms for the entire image. These models offer higher accuracy and are more resource-efficient, making them suitable for real-time implementation in resources constrained environments.

A lightweight, stable, and high precision apple leaf disease classification model using MobileNet⁴⁵ was introduced in⁴⁶. The main limitation of the approach is that the authors focused exclusively on a dataset containing only two common types of apple leaf diseases. Additionally, the technique achieved a relatively low accuracy of only 73.5%, suggesting room for improvement in both dataset diversity and model performance.

In¹⁰, the authors utilized DenseNet-121 as the fundamental model and modified the fully connected layer to classify six unique apple leaf diseases. In their work, the authors employed regression, multi-label categorization, and an attention loss function. Their model achieved accuracies of 93.51%, 93.31%, and 93.71% using regression, multi-label classification, and attention loss function approaches, respectively. However, the technique faced scalability challenges due to the high memory and processing requirements of DenseNet-121.

A region-of-interest-aware deep CNN was proposed in⁴⁷. The novelty of the method lies in the leaf spot attention method, which combines a future segmentation network and a spot-aware classification network to improve the discriminative power for identifying leaf diseases. Additionally, a hybrid contrast stretching technique to improve the distinguishability of diseased patches in apple leaves was proposed in¹². This study employed a modified masked region-based CNN (Mask RCNN) technique to segment spots, extracting features by transfer learning, and integrating Kapur entropy with SVM for selecting features. Such advanced architectures, while achieving high accuracy, often come with increased computational demands, making them less suitable for on-field applications without specialized hardware.

In¹³, Luo *et.al* proposed an improved version of the residual network (ResNet) model⁴⁸ to address the issue of information degradation in conventional ResNet models. Their model effectively improved the efficiency of classifying apple leaf diseases by segregating channel projection and spatial projection. To enhance the performance, they replaced the 3x3 convolution layer with pyramid convolution. The optimized model resulted in classification accuracy of 94.24% on the original dataset and 94.99% on the preprocessed dataset.

In¹⁴, a DL method based on an innovative multi-scale dense classification network was proposed to analyze and identify several categories of images, encompassing both healthy and damaged apple fruits and leaves. Their network model exceeded the classification accuracy of DenseNet-121. To enhance the accuracy of diagnosis, the researchers employed the Cycle-GAN algorithm to augment the dataset by generating anthracnose and ring rot lesions on healthy apple fruits. The multi-scale dense network design exhibited remarkable effectiveness in diagnosing a broad spectrum of apple diseases. However, the use of augmentation techniques, such as Cycle-GAN, while improving performance, may lead to overfitting to specific dataset characteristics, reducing robustness on unseen data.

For an imbalanced dataset,¹⁵ demonstrated that the lightweight CNN model, RegNet¹⁶, could accurately classify apple leaf diseases. Their model attained an impressive overall accuracy of 99.23%. To avoid overfitting, data augmentation of the training set and validation sets was carried out. However, the resulting data distributions were not balanced and the model performance was not reflective of a balanced dataset.

In², a DL ensemble model was proposed, integrating pre-trained DenseNet-121⁴⁹, EfficientNet-B7²⁶, and EfficientNet noisy student¹⁷ to enhance the accuracy of classifying apple leaves with different diseases. The authors utilized picture augmentation techniques, including Canny edge detection, flipping, and blurring, to enlarge the dataset and strengthen the model's resilience, leading to enhanced performance. The proposed model achieved a validation accuracy of 96.25%, surpassing several previous models. Their model demonstrated a high level of precision, reliably identifying apple leaves affected by different diseases with an accuracy rate of 90%. However, ensemble models like these inherently involve high memory and processing requirements, making them unsuitable for implementation in resource constrained environments.

EfficientNet-MG, a compact CNN, was proposed in¹⁸ for efficient classification of apple leaf diseases. The accuracy of the standard EfficientNet model was significantly enhanced by using the multi-stage feature fusion (MSFF) and Gaussian error linear unit (GELU) activation functions, resulting in an impressive accuracy of 99.11%. Models like EfficientNet-MG and A-Net, while highly accurate, often require advanced hardware for training and deployment, increasing the cost of implementation.

Furthermore,¹⁹ proposed an efficient sailfish optimizer (ESO) that uses an EfficientNet-based model to accurately identify and classify apple leaf diseases. The findings demonstrated the superiority of the proposed EfficientNet based apple leaf disease detection (ESFO-EALD) approach compared to existing methods.

In²⁰, an enhanced EfficientNet-v2 model was employed to classify seven prevalent apple leaf diseases. The model attained a remarkable accuracy of 97.49% in the model training recognition, boosted accuracy in recognizing complex scenes, and improved model parameters and training time. However, these advancements came at the cost of much higher computational requirements compared to the original Efficientnet (B0-B7) family.

While aforementioned DL techniques have significantly advanced the classification of apple leaf diseases, challenges such as high computational demands, limited dataset diversity, and the complexity of certain models persist. Addressing these limitations is essential for developing high performance, lightweight and scalable solutions that balance accuracy, efficiency, and practicality in real-world agricultural applications.

A number of recent studies show that there are growing efforts in the literature to address these challenges and develop DL classification architectures that excel in specific applications while addressing the need for developing high-performance, lightweight and scalable solutions. For example:

- EfficientNet models²⁶, especially EfficientNet-B0, offer an excellent balance between model complexity and performance, making them suitable for resource-constrained environments.
- Recent models such as PDNet⁵⁰ and CLDA-Net⁵¹ focus on reducing computational demands while improving accuracy through techniques like adapter blocks in PDNet and the convolutional block attention module (CBAM) in CLDA-Net.
- In one study⁵², researchers tackled challenges like class imbalance and negative transfer learning by employing statistical methods and stepwise transfer learning, and proposed a memory-efficient model suitable for resource-constrained devices.
- Another study⁵³, replaced standard convolution in Inception-v3 and InceptionResNet-V2 with depth-wise separable convolution, which reduced the number of parameters by a large margin while maintaining performance and accuracy.
- A comprehensive review²³ of fruit and vegetable disease detection highlighted the need for models that balance accuracy and resource efficiency.

In this context, our study builds upon recent advancements by introducing a fine-tuned EfficientNet-B0 model specifically designed for apple leaf disease classification. Leveraging several key optimizations and a holistic training strategy, the model achieves high accuracy with minimal computational overhead, making it well-suited for resource-constrained environments. These enhancements establish our model as a practical and scalable solution for plant disease detection. By addressing the computational and generalization challenges of existing methods, our approach contributes to the wider adoption of DL solutions in smart farming.

Methods

This section explains our stepwise methodology to develop the proposed fine-tuned EfficientNet-B0 model.

EfficientNet-B0 selection

Table 1 (Page 6) presents a detailed comparison of the accuracy, memory size, and FLOPs requirements of different EfficientNet variants and various CNNs on the ImageNet dataset^{26,54}. A close examination of Table 1 shows that EfficientNet variants achieve high accuracy with fewer parameters and lower FLOPs than other models. For instance, EfficientNet-B0 demonstrates 77.1% accuracy while maintaining a compact model size of 5.3 million parameters and a low computing requirement of 0.39 billion FLOPs. Although it sacrifices some accuracy compared to larger EfficientNet variants (B1-B7) and other CNN models, its efficiency in memory and computation makes it ideal for resource-constrained environments.

EfficientNet-B0's architecture is consistent across the series, making it suitable for large datasets under constrained computational resources. Its intermediate accuracy on ImageNet reflects its ability to capture critical features efficiently, making it highly applicable to tasks such as leaf disease classification. While larger variants offer slightly higher accuracy, their significant memory and computational demands make them less practical for real-time applications, particularly in agriculture.

This study leverages EfficientNet-B0's compact design by applying transfer learning and fine-tuning approaches for classifying apple leaf diseases. This ensures the model maintains computational efficiency while achieving high precision, recall, and F1 scores on domain-specific datasets, such as the APV and PV datasets. The choice of EfficientNet-B0 thus aligns with the study's objectives to develop a resource-efficient, robust and accurate disease classification model for practical agricultural applications.

Datasets

The PV dataset is an extensive open dataset consisting of 55,448 images showcasing a wide range of plant leaf diseases found in 14 different types of fruit and vegetable crops, which includes apple. The dataset comprises 38 distinct categories, including 17 fungal diseases, 4 bacterial diseases, 2 mycoses, 2 viral diseases, 1 disease caused by mites, and 12 categories indicating healthy crop leaves. In addition to the total of 55,448 images, the dataset has an extra class consisting of 1143 backdrop images. Figure 1 (Page 7) displays the spread of the leaf disease classes. Several research publications have employed the PV dataset to assess the effectiveness of DL models in detecting plant diseases^{55–59}.

Model	Top-1 Acc.	#Params	#FLOPs (Approx)
EfficientNet-B0	77.1%	5.3 M	0.39 B
EfficientNet-B1	79.1%	7.8 M	0.69 B
EfficientNet-B2	80.1%	9.2 M	1.00 B
EfficientNet-B3	81.6%	12 M	1.80 B
EfficientNet-B4	82.9%	19 M	4.20 B
EfficientNet-B5	83.6%	30 M	10.3 B
EfficientNet-B6	84.0%	43 M	19.0 B
EfficientNet-B7	84.3%	66 M	37.0 B
ResNet-50	76.0%	26 M	4.1 B
DenseNet-169	76.2%	14 M	3.5 B
ResNet-152	77.8%	60 M	11 B
DenseNet-264	77.9%	34 M	6.0 B
Inception-v3	78.8%	24 M	5.7 B
Xception	79.0%	23 M	8.4 B
Inception-v4	80.0%	48 M	13 B
Inception-resnet-v2	80.1%	56 M	13 B
ResNeXt-101	80.9%	84 M	32 B
PolyNet	81.3%	92 M	35 B
SENet	82.7%	146 M	42 B
NASNet-A	82.7%	89 M	24 B
AmoebaNet-A	82.8%	87 M	23 B
PNASNet	82.9%	86 M	23 B
AmoebaNet-C	83.5%	155 M	41 B
GPipe	84.3%	557 M	-

Table 1. Performance results on ImageNet for EfficientNet variants and different variants.(Source:²⁶).

To classify apple leaf diseases, both the PV and APV datasets were utilized. The APV dataset consists of 3,171 healthy and diseased apple leaf images. It is comprised of 51.87% healthy leaf images, 19.86% apple scab images, 19.58% black rot images, and 8.67% apple cedar rust images.

Figure 2 (Page 7) displays examples of both healthy and diseased apple leaf images from the APV/PV dataset. Figure 2a clearly illustrates that healthy leaves exhibit a pristine appearance, devoid of any blemishes, and possess a vibrant green color, indicating the absence of any disease. Figure 2b displays a leaf from an apple tree that has brown patches or marks, which are indicative of a scab infection. The scab infection is mostly caused by the fungal pathogen *Venturia inaequalis*, which commonly infects both the leaves and fruits of plants, resulting in the fruit becoming unfit for consumption. Figure 2c depicts an apple leaf exhibiting symptoms of black rot. Black rot is identified by the presence of dark, depressed lesions on fruits and foliage, which are caused by the fungus *Botryosphaeria obtusa*. In Fig. 2d, the leaf of an apple tree exhibits symptoms of cedar-apple rust, characterized by dense yellowish markings. The etiology of this condition is attributed to the fungal pathogen *Gymnosporangium juniperi-virginianae*.

Data pre-processing

The dataset images were resized from 256×256 to 224×224 to meet the input size specifications of CNN models. Also, the pixel values underwent normalization. This involved dividing each pixel value by the maximum possible pixel value, which is typically 255 for 8-bit per channel images such as RGB. It is crucial to note that normalization is essential for ensuring uniform pixel values across all images. Additionally, it promotes the stability of the model during the training process and enhances its capacity to make correct predictions, leading to enhanced generalization.

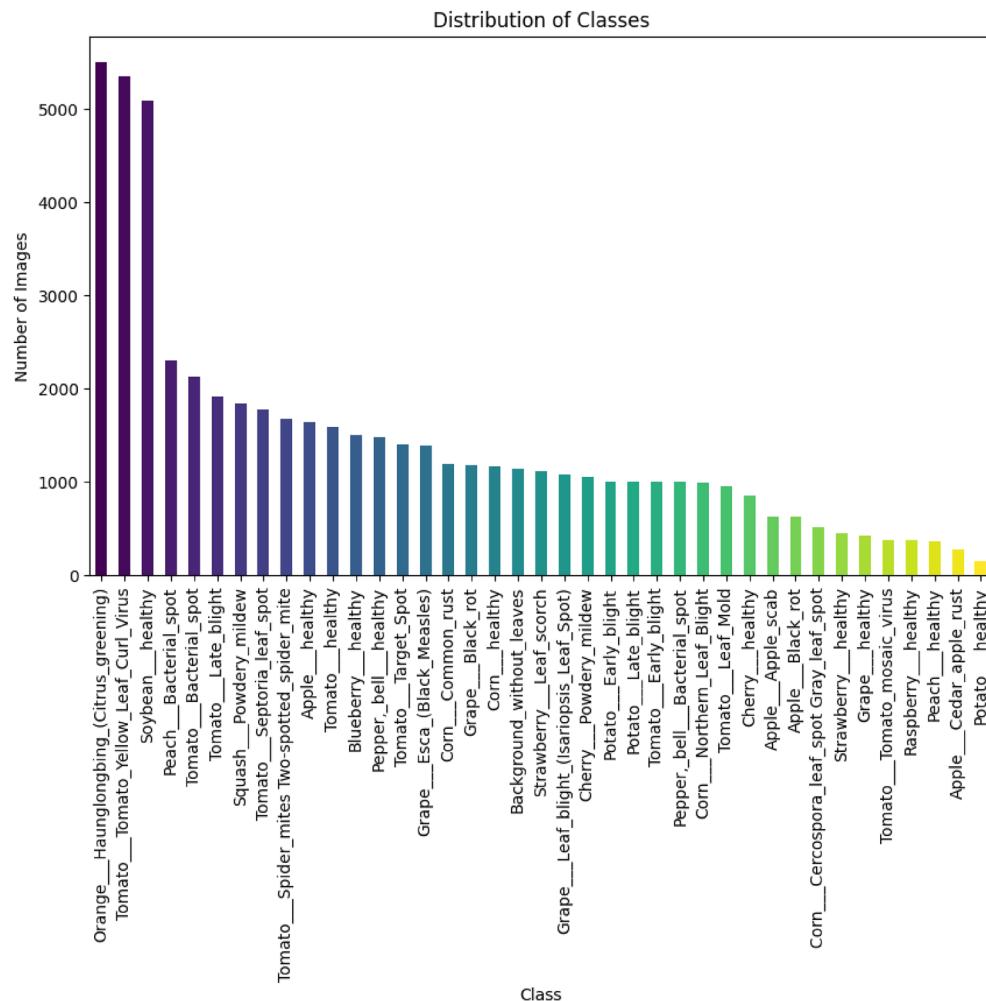
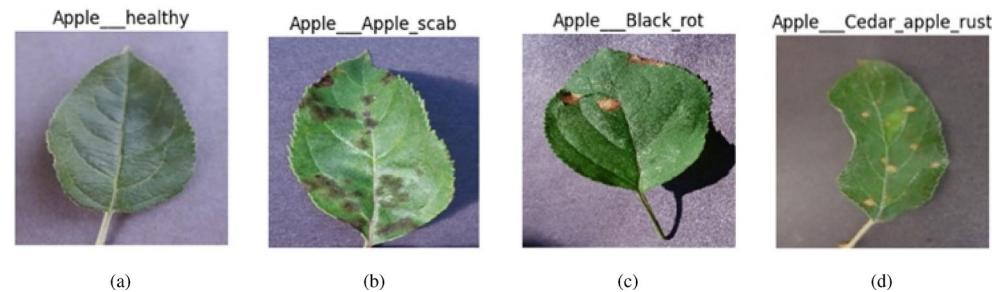
Data partitioning

To enhance the model training and evaluation efficiency, an 80-20 data split was implemented on both the datasets. 80% of each dataset was assigned to the training set, while the remaining 20% was evenly split between the validation and test sets. Table 2 (Page 7) displays the distribution of data across the datasets after the split.

Data balancing

Both the APV and PV datasets exhibit an uneven distribution of classes, which requires taking steps to ensure that CNN models can effectively generalize across all classes without favoring the majority class.

To address class imbalance in the datasets, stratified sampling^{31,32} was employed during the data partitioning process. Stratification ensures that the proportion of samples in each class remains consistent across the training, validation, and test subsets. This approach helps preserve the representativeness of the dataset, preventing over-representation or under-representation of specific classes that could lead to biased model training and evaluation.

**Fig. 1.** Distribution of classes in the PV dataset.**Fig. 2.** Samples of healthy and unhealthy apple leaf images from the PV/APV dataset: (a) healthy, (b) apple scab, (c) black rot, (d) cedar apple rust.

Dataset	Total	Training data	Validation data	Test data
APV	3171	2536	317	318
PV	55,448	44,358	5545	5545

Table 2. Distribution of datasets after the 80–20% split.

Mathematically, let D represent the dataset with K images distributed among C classes. Let k_i denote the number of images in class i , where $i = 1, 2, \dots, C$, such that: $\sum_{i=1}^C k_i = K$. The proportion p_i of images in class i is defined as: $p_i = \frac{k_i}{K}$. For the stratified partitioning, if D_{train} , D_{val} , and D_{test} represent the training, validation, and test subsets with sizes K_{train} , K_{val} , and K_{test} , respectively, the number of images $k_{i,\text{train}}$, $k_{i,\text{val}}$, and $k_{i,\text{test}}$ in each subset for class i is calculated as: $k_{i,\text{train}} = p_i \cdot K_{\text{train}}$, $k_{i,\text{val}} = p_i \cdot K_{\text{val}}$, $k_{i,\text{test}} = p_i \cdot K_{\text{test}}$. By maintaining these proportions, stratified sampling ensures that the class distribution in each subset mirrors the original dataset, preserving balance across all partitions.

This methodology was implemented using the `StratifiedShuffleSplit` function from the `scikit-learn` library. This stratified partitioning not only ensures fair representation of all classes but also enhances the model's ability to generalize to unseen data, which is particularly critical when working with imbalanced datasets.

Data augmentation is a frequently employed technique in ML and DL that artificially expands the size of a dataset by applying different changes to the current data samples^{29,30}. To address the issues arising from imbalanced datasets and the risk of overfitting, data augmentation was applied on the 80% training subset after the initial split, specifically for our datasets.

To implement data augmentation, the `Keras` package was used to apply a range of transformations to the datasets. These changes included random rotations, horizontal flips, zooming within a range of 20%, rotation within a range of 30 degrees, and brightness adjustments between the range of 0.8 to 1.2. This strategic approach increased the number of images in each class, creating a more balanced dataset and successfully reducing the risk of overfitting in the model. Figure 3 (Page 9) displays samples of healthy and diseased apple leaf images from the PV/APV dataset after data augmentation.

The class weighting approach³³ was employed to tackle the data imbalance problem. By calculating class weights, our model effectively assigns the right level of importance to the learning of each class. The model's capacity to recognize patterns in classes with fewer images was improved by making proportional adjustments during the training phase. This resulted in a robust and unbiased model that could effectively handle imbalanced datasets.

The weight for the class i (CW_i) was computed as:

$$CW_i = \frac{K}{C \cdot k_i} = \frac{1}{C \cdot p_i}. \quad (1)$$

The weights of each class were calculated using the `scikit-learn` function `compute_class_weights`³³. These weights were then transferred to the `model.fit` training function in TensorFlow.

Transfer learning and fine-tuning

Transfer learning is a very effective technique that utilizes existing features and model parameters/weights from one assignment to improve performance on a new task. This approach reduces the time required for training and enhances classification accuracy by utilizing pre-trained models^{60,61}. On the other hand, fine-tuning is a precise method used in transfer learning. It involves training a pre-trained model, which has already been trained on a vast dataset such as ImageNet, on a fresh dataset to enhance its performance. This strategy utilizes the acquired knowledge in the model (network) by employing various combinations of pre-trained layers: freezing certain layers, fine-tuning others, and adding new layers⁶¹. Both pre-trained models and fine-tuning have been effectively utilized in much research for the classification of plant leaf diseases^{62–65}.

Due to the limited quantity and characteristics of our training datasets for DNNs, it has become necessary to use pre-trained weights from ImageNet. The classification layers (the top levels of the pre-trained model) were replaced with new layers and all the model parameters were fine-tuned rather than keeping some of them fixed. This comprehensive fine-tuning strategy was essential to attaining the highest level of accuracy during the model development. By fine-tuning the learned representations from ImageNet to align with the features of the APV and PV datasets, important features related to plant diseases were identified, resulting in improved classification performance.

Model architecture

The EfficientNet-B0 base model architecture, shown in Fig. 4 (Page 9), comprises two main components: the backbone and the head. The backbone consists of convolutional layers and MobileNetV2 inverted residual bottleneck convolution (MBConv) blocks. The purpose of the backbone is to extract features from input images. Conversely, the head section of the model, comprising the GAP layer, fully connected layers, and the output layer, utilizes these features to generate the final predictions for the image classification task.

The following is a comprehensive discussion of the architecture of EfficientNet-B0:

- **Input layer:** The input layer receives the input images and passes them to the network model. The input images are resized to dimensions of either 224x224 pixels or, less frequently, 260 × 260 pixels.
- **Stem convolution:** Each input image is subjected to a sequence of convolutional procedures to extract low-level features from the input images. Usually, the input image undergoes a series of 3×3 convolutions.
- **EfficientNet blocks:** EfficientNet-B0 is composed of several blocks, each including a series of MBConv layers. MBConv layers are the fundamental components of EfficientNet, integrating inverted residual connections with depth-wise separable convolutions. Every MBConv block has the task of extracting and improving features at various levels of abstraction.
- **GAP layer:** Following the convolutional layers, a GAP layer is used to calculate the average value of each feature map, essentially condensing the spatial information into a singular vector.

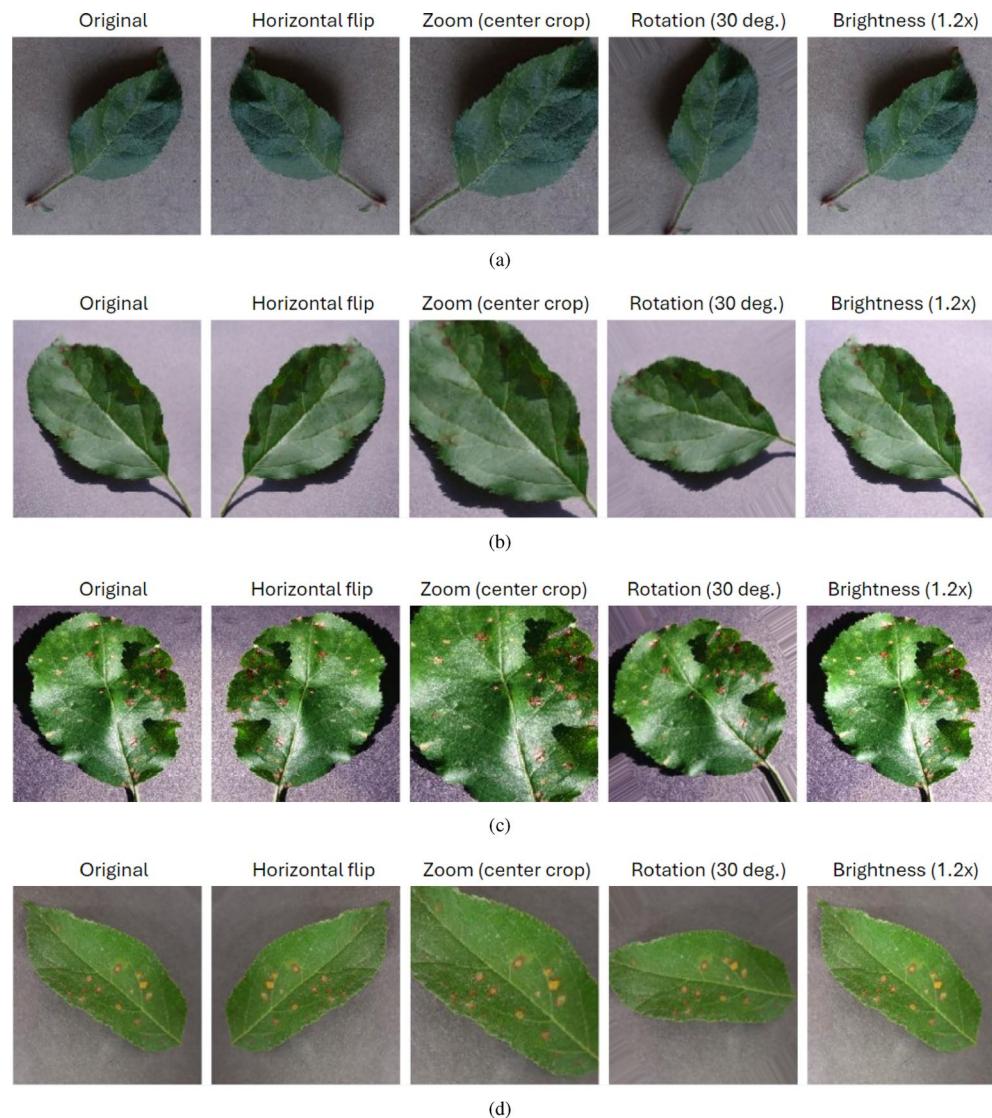


Fig. 3. Samples of healthy and diseased apple leaf images from the PV/APV dataset after data augmentation: (a) augmented transformations of a healthy apple leaf, (b) augmented transformations of an apple scab leaf, (c) augmented transformations of a black rot leaf, and (d) augmented transformations of a cedar apple rust leaf.

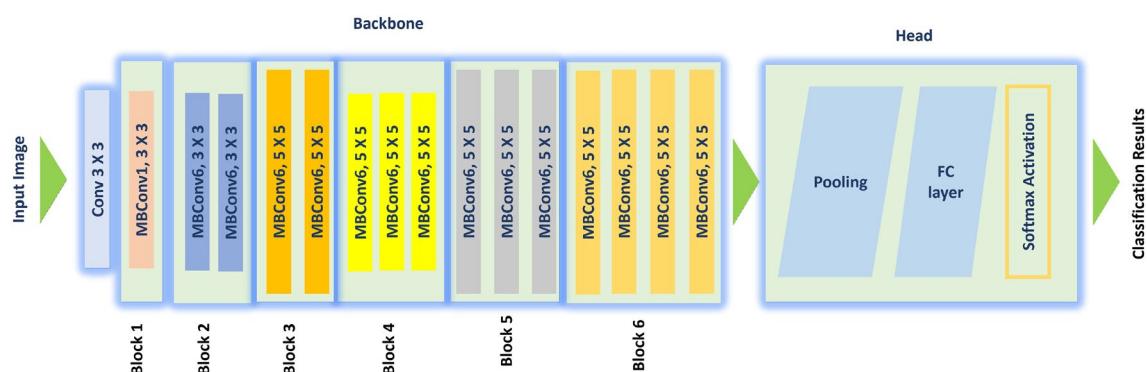


Fig. 4. Architecture of EfficientNet-B0 with MBConv as basic building blocks.

- **Fully connected layers:** After the GAP layer, there are usually one or more fully connected layers. The purpose of these layers is to transform the high-level features obtained from the convolutional layers into the required output classes.

The output layer generates the final classification probabilities for the given input image. The last layer often has a Softmax activation function, which transforms the unprocessed model output into probabilities for each class.

To converge faster and improve the performance, the pre-trained EfficientNet-B0 model was initialized using its learned parameters obtained from the ImageNet dataset. Also, to adapt the model architecture for our specific application and further improve the performance, modifications were made to the last layers of the classification section (known as the head). These modifications involved replacing the existing head model with a new model consisting of six layers.

1. A GAP layer was used in the original EfficientNet-B0 architecture to reduce the spatial dimensions of feature maps by averaging the activations across each feature map. However, for the specific task of detecting localized patterns such as disease spots or lesions on apple leaves, GAP layer was replaced with GMP layer^{66,67} as the first layer.

GMP computes the maximum activation across the entire spatial dimensions of each feature map. Given a feature map F of dimensions $H \times W$, where H and W are the height and width, respectively, GMP is defined as:

$$\text{GMP}(F) = \max_{i,j} f_{ij}, \quad (2)$$

where f_{ij} represents the activation at position (i, j) . For a multi-channel feature map, GMP is computed independently for each channel, resulting in a single scalar value per feature map.

This modification was inspired by GMP's ability to focus on the most salient features in the input, making it particularly effective in identifying localized patterns critical for leaf disease identification. In order to prevent the over-smoothing effect of GAP, which might dilute key localized characteristics by averaging, GMP prioritizes the most activated locations in the feature maps, ensuring that substantial information is retained.

2. The second layer consists of a batch normalization layer⁶⁸, which is designed to enhance the speed, efficiency, and reliability of our model. Batch normalization addresses the problem of internal covariate shift (CS) in deep neural network training by normalizing the input of each layer. This process adjusts the mean and variance, which enables the use of greater learning rates and simplifies initialization⁶⁹.
3. A dense layer consisting of 256 nodes is implemented after the batch normalization layer. The rectified linear units (ReLU) activation function is defined as $\text{ReLU}(x) = \max(0, x)$, where x denotes the input value. ReLU introduces non-linearity by setting negative inputs to zero and passing positive inputs unchanged. Object classification is facilitated in fully connected layers by using nonlinear ReLU activation functions⁷⁰. ReLU activation was implemented on the fully connected layer to enhance the non-linearity in the images.
4. L1 and L2 regularization techniques are used to mitigate overfitting in DL models⁷¹. L1 regularization promotes sparsity by penalizing the absolute values of weights, but L2 regularization penalizes the squared magnitudes of weights, hence encouraging smaller and more uniformly distributed weights across all features. After applying the ReLU activation layer, a layer of L1 and L2 regularization was incorporated into the network to mitigate overfitting.
5. The dropout⁷² technique can be used to mitigate the risk of model overfitting. It randomly excludes neurons and their connections during the training process. This approach operates by randomly sampling from various "thinned" networks, which introduces diversity during the training process⁷³. By preventing co-adaptation difficulties and weight updates during the backward pass, the network's sensitivity to the weight of individual neurons is reduced. Consequently, a dropout layer was included in the model after the L1 and L2 regularization layer. A dropout rate of 20% was employed, indicating that there is a 20% chance for each hidden unit to be assigned a value of 0 throughout the training process.
6. The output layer of the model utilizes a SoftMax activation⁷⁴ for multi-class classification. This activation allows the model to assign confidence probabilities to each class in the multi-class classification task. The predicted class is determined by selecting the class with the highest probability, providing both the predicted class and a measure of the model's confidence in the selection.

Both the APV and PV datasets exhibit substantial dissimilarities compared to the ImageNet dataset that was utilized to pre-train EfficientNet-B0. It was thus decided to fine-tune the complete model, including both the upper and lower layers. This enables the model to adjust more efficiently to the new task by acquiring task-specific characteristics while still leveraging the knowledge stored in the pre-trained model. During the training of our model, all layers were set to be trainable, resulting in the updating of weights for pre-trained layers as well. Subsequently, the accuracy and effectiveness of the fine-tuned model were verified through validation data.

Figure 5 (Page 11) illustrates the process of transfer learning and fine-tuning the Efficient-Net-B0 model for the APV/PV dataset.

Loss function

Categorical cross-entropy (CCE)⁷⁵ was used as the selected loss function. The calculation of CCE for several classes on a complete data set containing n records can be expressed as:

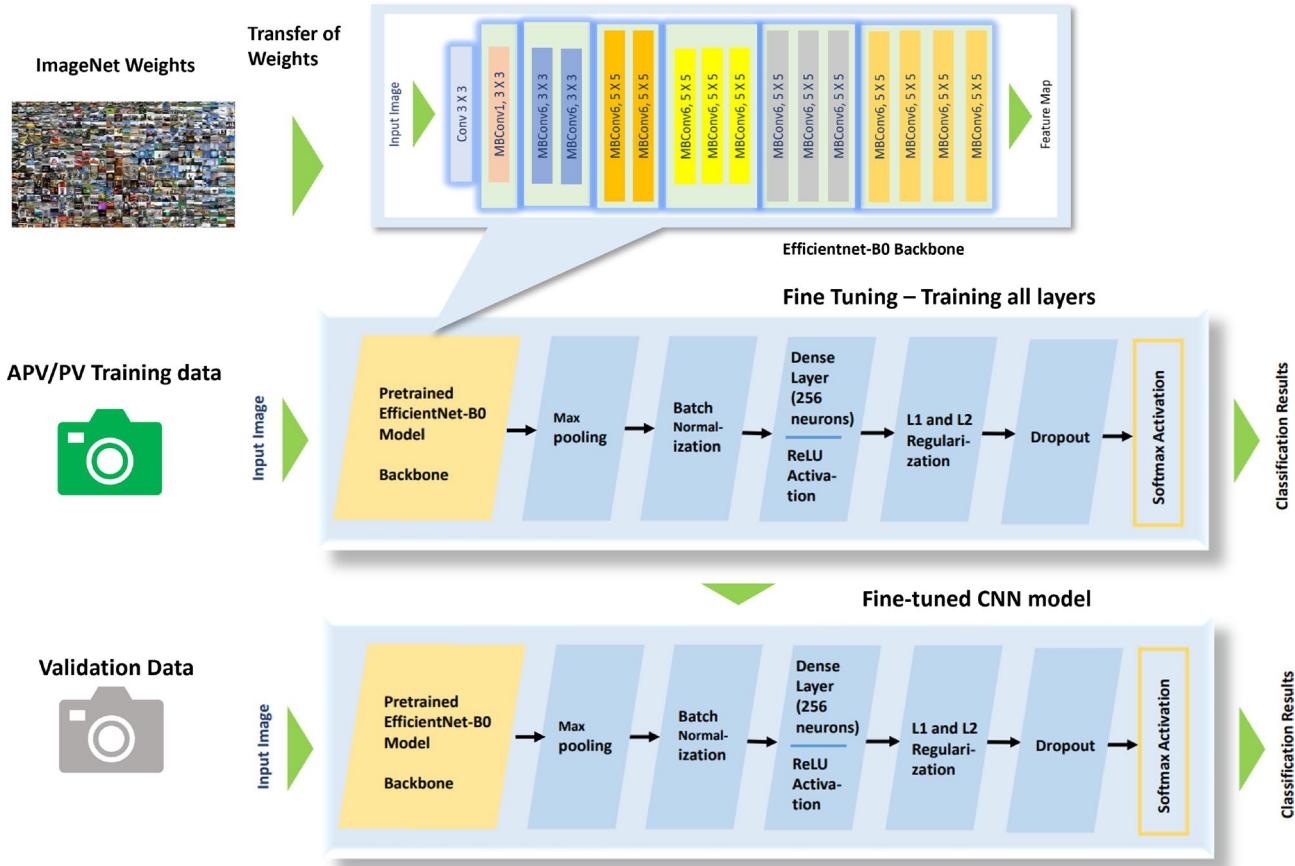


Fig. 5. Transfer learning and fine-tuning process of the EfficientNet-B0 for the APV/PV dataset.

$$\text{Categorical cross entropy} = - \sum_{i=1}^n \left(\sum_{j=1}^m y_j \log(\hat{y}_j) \right), \quad (3)$$

where $i = 1$ to n records and $j = 1$ to m classes. The choice of CCE is in line with the multi-class characteristic of the plant disease classification task. It penalizes discrepancies between anticipated and actual class probabilities and encourages precise class differentiation.

Optimizer

The AdaMax optimizer^{76,77} was used during model compilation. The decision to choose AdaMax was based on its appropriateness for situations that involve regularization techniques such as L1 and L2, which result in sparsity in the model's parameters. Its selection was further supported by the comparative optimizer analysis presented in the "Discussion" section.

The learning rate exerts a significant impact. A learning rate of 0.001 is typically employed as a standard practice to provide rapid training and convergence to a good solution⁷⁸. Therefore, the AdaMax optimizer was used with a learning rate of 0.001 during the model compilation process.

Evaluation metrics

A confusion matrix is a commonly used tool to visually represent the accuracy of a trained model in predicting outcomes from a validation dataset. This research thoroughly evaluates the model's performance by utilizing precision, recall, accuracy, and F1-Score metrics. These metrics are calculated based on the values of true positives (T_P), true negatives (T_N), false positives (F_P), and false negatives (F_N) obtained from the confusion matrix. The metrics are outlined in the following: *Accuracy* metric quantifies the ratio of correctly identified image samples to the total number of image samples. The calculation involves adding the number of true positives and true negatives and then dividing the result by the total number of image samples.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}. \quad (4)$$

Precision metric quantifies the ratio of true positive predictions to all positive predictions. The calculation involves dividing the number of true positives by the sum of true positives and false positives.

$$\text{Precision} = \frac{T_P}{T_P + F_P}. \quad (5)$$

Recall, also known as sensitivity, is the ratio of correctly predicted positive samples to the total number of actual positive samples. The calculation involves dividing the number of true positives by the sum of true positives and false negatives.

$$\text{Recall} = \frac{T_P}{T_P + F_N}. \quad (6)$$

The *F1 score* is calculated as the harmonic mean of precision and recall. It achieves a balance between accuracy and recall.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (7)$$

Experimental setup

The experimental configuration involved a MacBook Pro using a 2.3 GHz Quad-Core Intel Core i5 processor, supported by Intel Iris Plus Graphics 655 with 1536 MB, and 8 GB of 2133 MHz LPDDR3 RAM. The coding operations were executed using Google Colab, utilizing TensorFlow and Keras libraries to maximize the utilization of the GPU resources for faster calculation. The training regimen consisted of 20 epochs, with a learning rate of 0.001 and a batch size of 32.

Results

Training and validation performance

Figure 6a and b (Page 13) present the training and validation accuracy curves, and the cross-entropy loss curves, respectively, obtained using the APV dataset for both the fine-tuned EfficientNet-B0 and EfficientNet-B0 models.

The results show a steady and continuous decrease in loss for the fine-tuned EfficientNet-B0, indicating a stable learning process. Furthermore, there are no noticeable indications of overfitting or under-fitting. On the other hand, the learning process of EfficientNet-B0 demonstrates a lack of stability. It also needs a greater number of epochs to minimize the difference between the training and validation curves.

The training accuracy for both models is seen to quickly increase in the first 5 epochs and steadily increase over the next 5 epochs. There is no further appreciable rise in training accuracy as the number of epochs increases. The validation accuracies for both models exhibit a continuous and increasing pattern over epochs, suggesting successful generalization to new and unseen data. The validation curve of the fine-tuned EfficientNet-B0 closely matches its training curve around the 17th epoch. However, the validation progress of EfficientNet-B0 seems to be slower and less consistent.

Fine-tuned EfficientNet-B0 exhibits a steady decline in training losses over the epochs, indicating a constant enhancement in predicting the training data. In contrast, EfficientNet-B0 depicts an extremely low training loss in the first few epochs. This may not always be a problem but could be a sign of a potential overfitting problem. Also, the validation losses for both models reduce over epochs, the trend is slow and less consistent in the case of EfficientNet-B0.

Figure 7a and b (Page 14) display the training and validation accuracy curves, and the cross-entropy loss curves, respectively, for both the fine-tuned EfficientNet-B0 and EfficientNet-B0 models using the PV dataset. Consistent training and validation progress for the fine-tuned EfficientNet-B0 model is witnessed. On the other hand, the validation accuracy and loss curves for EfficientNet-B0 display erratic patterns, with abrupt spikes and drops, suggesting overfitting to the training data.

Classification performance

To evaluate the model's effectiveness in detecting apple leaf diseases, we utilized the APV dataset, which includes only apple-specific leaf disease classes for training and testing. The confusion matrix for the fine-tuned EfficientNet-B0 model on the APV test dataset is presented in Fig. 8 (Page 15). The projected categories can be seen to align closely with the true classifications, with the model accurately classifying most apple leaf images with high probability. Specifically, out of the 28 sample images in the “apple cedar rust” class, all but one are correctly classified. The remaining three groups were accurately classified. As a result, the fine-tuned model correctly classifies 317 out of a total of 318 images.

Table 3 (Page 15) shows the fine-tuned model's precision, recall, and F1 score metrics (derived from the APV dataset confusion matrix). The precision values, ranging from 0.98 to 1.0, suggest that the model's predictions for each disease class are extremely accurate, with very few false positives. Recall values ranging from 0.96 to 1.0 indicate that the model is highly effective at identifying most of the positive events for each disease class. In addition, the F1 scores, ranging from 0.98 to 1.0, highlight the model's capacity to create a balanced trade-off between precision and recall. Overall, these results indicate that the fine-tuned model has a strong ability to recognize apple leaf diseases accurately and reliably.

To assess the model's generalizability, we used the broader PV dataset, which includes 39 plant disease and healthy categories. The confusion matrix for this dataset is shown in Fig. 9 (Page 16). The results show that predicted categories align with the true categories of the test images, and the model successfully and accurately

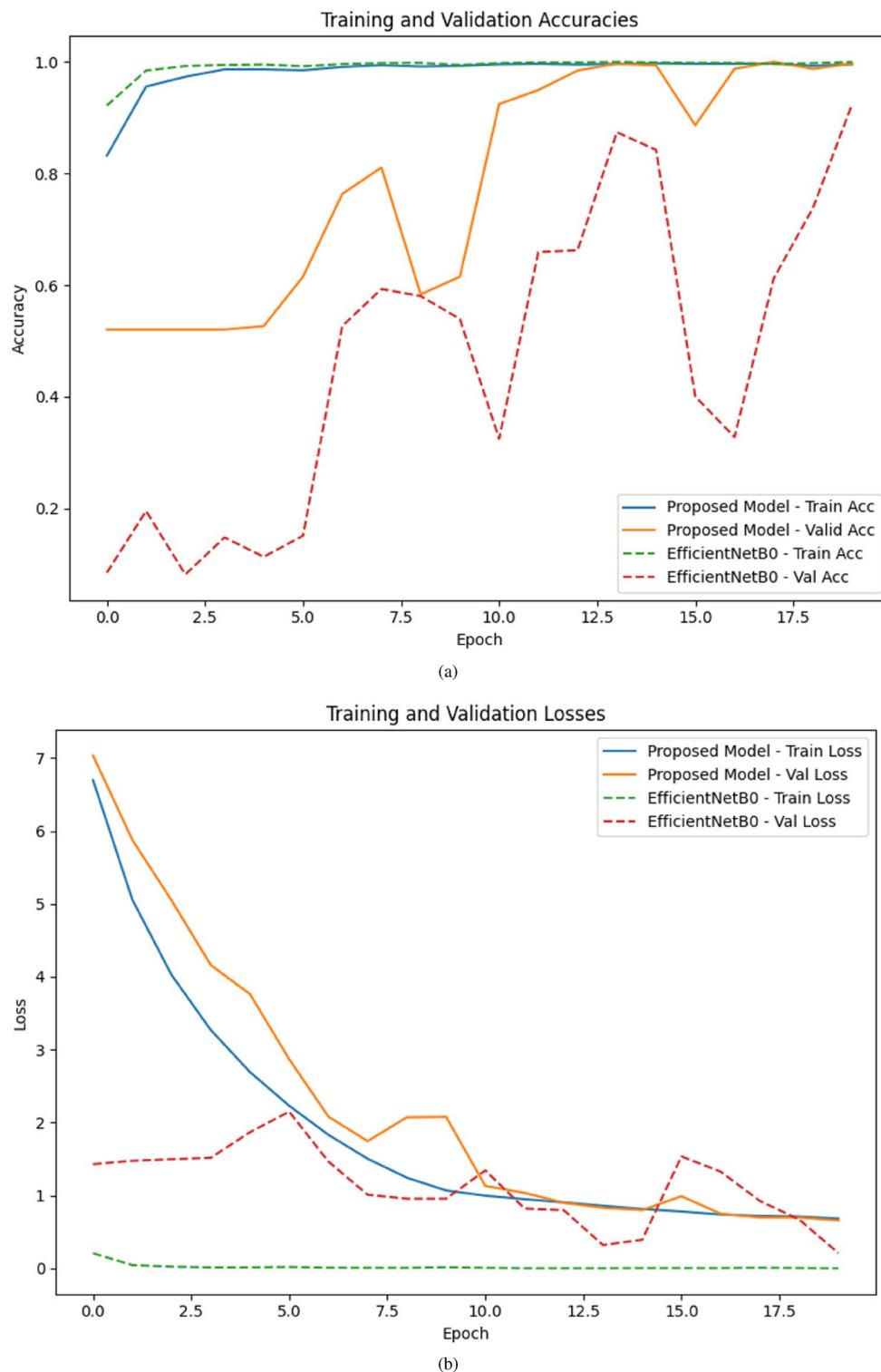


Fig. 6. Performance results for fine-tuned EfficientNet-B0 and EfficientNet-B0 on the APV dataset: (a) training and validation accuracy curves, (b) training and validation loss curves.

differentiates between a variety of plant diseases, including apple-specific categories like apple scab, black rot, and cedar-apple rust, with high accuracy. This evaluation underscores the model's robustness and scalability, essential for practical agricultural applications.

Table 4 (Page 17) provides a detailed breakdown of the model's performance metrics derived from the PV dataset confusion matrix. It highlights consistently high precision, recall, and F1 scores across all 39 classes (including apple-specific categories), often achieving perfect scores of 1.0. This analysis highlights the model's

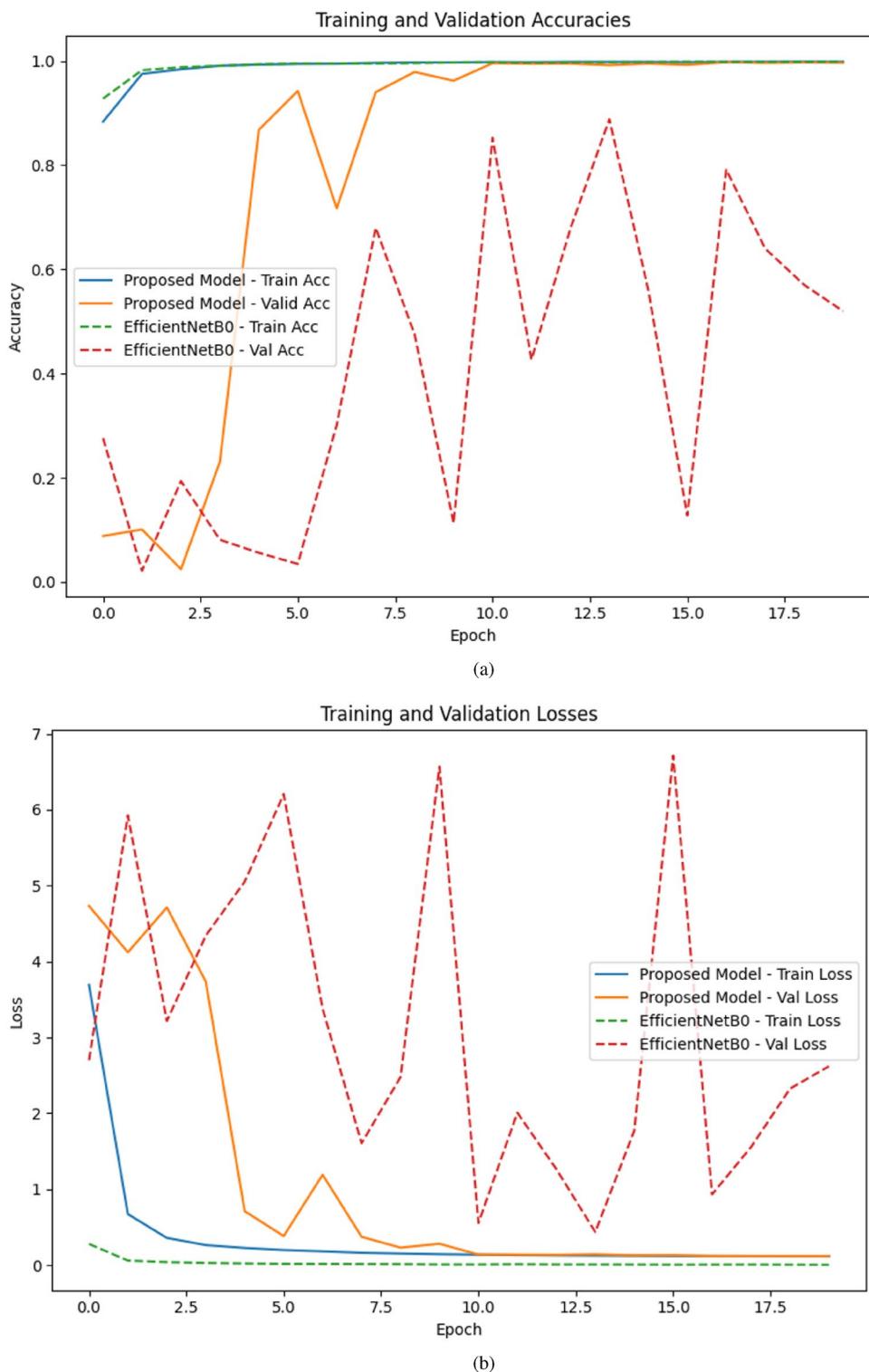


Fig. 7. Performance results for fine-tuned EfficientNet-B0 and EfficientNet-B0 on the PV dataset: (a) training and validation accuracy curves, (b) training and validation loss curves.

reliability and versatility in managing multiple plant diseases simultaneously, a crucial aspect for implementation in agricultural environments.

This broader evaluation emphasizes key aspects of the fine-tuned EfficientNet-B0 model:

- **Robustness:** High accuracy across diverse plant disease categories, including apple-specific classes like apple scab, black rot, and cedar-apple rust, demonstrates its resilience in varied scenarios.

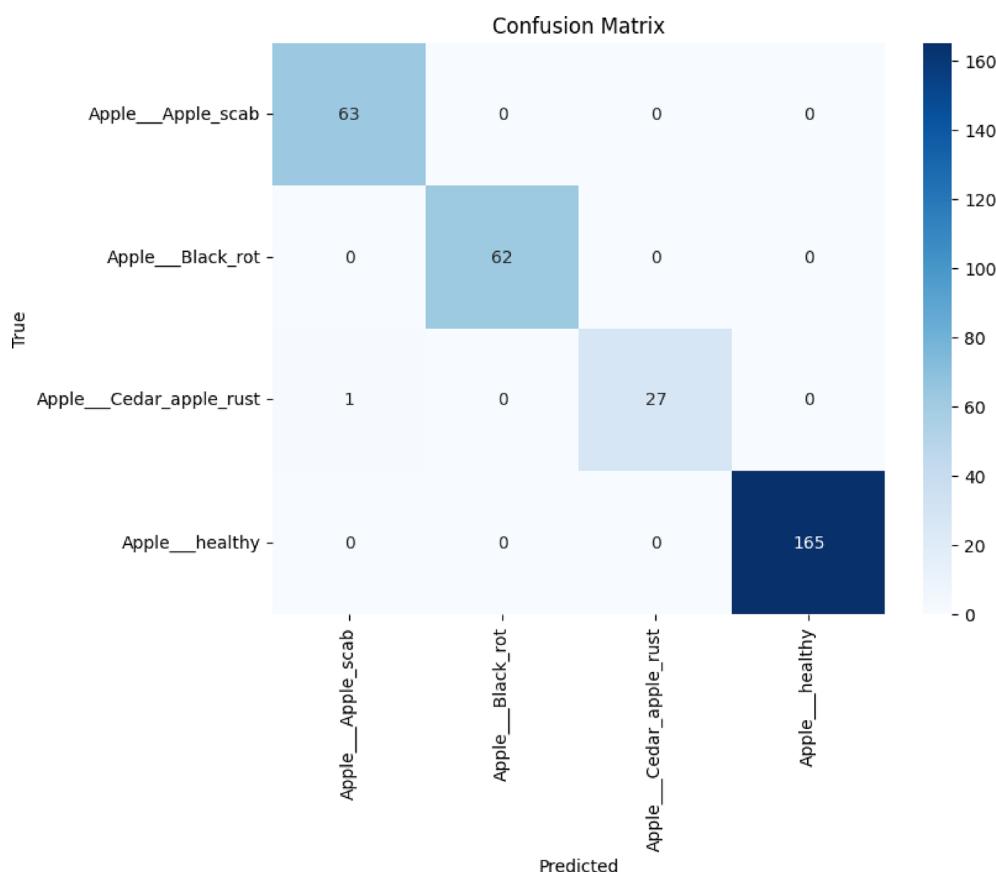


Fig. 8. Confusion matrix of fine-tuned EfficientNet-B0 on the APV test dataset.

Class/Category	Precision	Recall	F1 Score
Apple Scab	0.98	1.00	0.99
Black Rot	1.00	1.00	1.00
Cedar-Apple Rust	1.00	0.96	0.98
Healthy	1.00	1.00	1.00

Table 3. Performance metrics for disease classes using the APV dataset.

- **Scalability:** Consistent performance across the PV dataset indicates its potential for applications beyond a single crop or disease, making it adaptable to broader agricultural needs.
- **Practicality:** Its lightweight architecture and strong generalization capabilities make it well-suited for deployment in resource-constrained environments, supporting efficient and scalable disease management.

These findings affirm the model's exceptional performance in both specialized APV and generalized PV scenarios, reinforcing its robustness and making it a valuable tool for farmers and agricultural practitioners to effectively manage plant health.

Comparison with the other state-of-the-art models

A comparative analysis of the fine-tuned model with other state-of-the-art CNN models was performed. For a fair comparison, the same learning rate and the Adam optimizer were employed for these models.

Table 5 (Page 17) displays a performance evaluation of the fine-tuned model in contrast to other models on the APV dataset. The fine-tuned model can be seen to surpass the performance of EfficientNet-B0, EfficientNet-B3, and VGG16, attaining an accuracy, recall, and F1 score of 0.9969, while matching the performance of top models such as Inception-v3 and ResNet-50. In addition, the fine-tuned model achieves a remarkable accuracy of 99.69%, surpassing the accuracy of EfficientNet-B0, EfficientNet-B3, and VGG16, while also matching the accuracy of Inception-v3 and ResNet-50. The results demonstrate the higher performance of the fine-tuned EfficientNet-B0 model in classifying diseases.

Table 6 (Page 18) provides a detailed comparison of the fine-tuned model's performance using the PV dataset. A careful examination of Table 6 reveals that EfficientNet-B0 has a modest level of performance, as indicated by

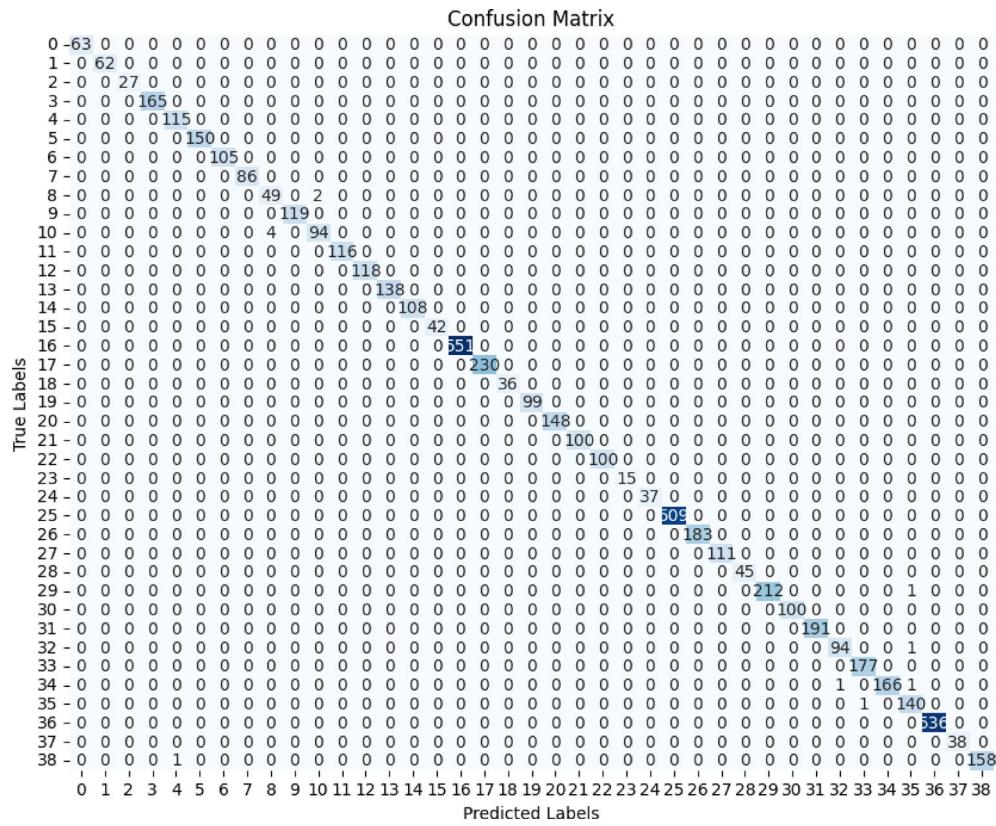


Fig. 9. Confusion matrix of fine-tuned EfficientNet-B0 using the PV test dataset, where 0-38 in the confusion matrix represents 39 classes in the PV dataset.

its precision, recall, and F1 score of 0.7057, 0.5028, and 0.4886, respectively. Also, it has a low accuracy of 50.28%. EfficientNet-B3 demonstrates improvement in all parameters, with precision, recall, and F1 score of 0.8822, 0.7399, and 0.7630, respectively, and an accuracy of 73.99%. Both Inception-v3 and ResNet-50 demonstrate outstanding performance, with precision, recall, and F1 score surpassing 0.99, and accuracies of 99.22% and 99.26% respectively. On the other hand, VGG16 exhibits poor performance in terms of precision, recall, and F1 scores, which are all nearly zero, and it achieves an accuracy of merely 1.5%. Remarkably, the fine-tuned EfficientNet-B0 outperforms all other state-of-the-art models. Boasting precision, recall, and F1 score of 0.9979, 0.9978, and 0.9978%, this model outperforms other models in classifying diverse diseases using the PV dataset.

Table 7 (Page 18) shows memory consumption in megabytes (MB) and inference speed in millions (M) of FLOPs, by different CNN models, with an input size of 224x224. The memory required, assuming the weights are stored as 32-bit floating-point values (4 bytes) and there are N weight parameters, can be computed as follows:

$$\text{Memory in MB} = \frac{N \times 4}{1024 \times 1024}. \quad (8)$$

Overall, both the EfficientNet-B0 and fine-tuned EfficientNet-B0 emerge as the most memory and FLOPs efficient choices among the comparative models. However, the fine-tuned EfficientNet-B0 achieves significantly higher accuracy than the EfficientNet-B0 model. Specifically, it yields an 11% increase in accuracy on the APV dataset and an impressive 49.5% increase in accuracy on the PV dataset. Surprisingly, this performance enhancement comes at a slight price of approximately a 7–8% increase in both the memory footprint and FLOPs when compared with the EfficientNet-B0. Consequently, the fine-tuned EfficientNet-B0 is seen to strike an

Class/Category	Precision	Recall	F1 Score
Apple Scab	1.00	1.00	1.00
Apple Black Rot	1.00	1.00	1.00
Cedar Apple Rust	1.00	1.00	1.00
Apple Healthy	1.00	1.00	1.00
Background without leaves	0.99	1.00	1.00
Berry Healthy	1.00	1.00	1.00
Cherry Powdery Mildew	1.00	1.00	1.00
Cherry Healthy	1.00	1.00	1.00
Corn Gray Leaf Spot	0.94	0.96	0.95
Corn Common Rust	1.00	1.00	1.00
Corn Northern Leaf Blight	0.98	0.97	0.97
Corn Healthy	1.00	1.00	1.00
Grape Black Rot	1.00	1.00	1.00
Grape Esca	1.00	1.00	1.00
Grape Leaf Blight	1.00	1.00	1.00
Grape Healthy	1.00	1.00	1.00
Citrus Greening	1.00	1.00	1.00
Peach Bacterial Spot	1.00	1.00	1.00
Peach Healthy	1.00	1.00	1.00
Bell Pepper Bacterial Spot	1.00	1.00	1.00
Bell Pepper Healthy	1.00	1.00	1.00
Potato Early Blight	1.00	1.00	1.00
Potato Late Blight	1.00	1.00	1.00
Potato Healthy	1.00	0.93	0.97
Raspberry Healthy	1.00	1.00	1.00
Soybean Healthy	1.00	1.00	1.00
Squash Powdery Mildew	1.00	1.00	1.00
Strawberry Leaf Scorch	1.00	1.00	1.00
Strawberry Healthy	1.00	1.00	1.00
Tomato Bacterial Spot	1.00	1.00	1.00
Tomato Early Blight	0.99	1.00	1.00
Tomato Late Blight	1.00	0.99	1.00
Tomato Leaf Mold	0.99	1.00	0.99
Tomato Septoria Leaf Spot	0.99	1.00	1.00
Tomato Spider Mites	1.00	0.99	1.00
Tomato Target Spot	1.00	0.99	1.00
Tomato Yellow Leaf Curl Virus	1.00	1.00	1.00
Tomato Mosaic Virus	1.00	1.00	1.00
Tomato Healthy	1.00	0.99	1.00

Table 4. Performance metrics for disease classes on the PV dataset.

Model	Precision	Recall	F1 Score	Accuracy (%)
EfficientNet-B0	0.8954	0.8868	0.8775	88.68
EfficientNet-B3	0.9256	0.9182	0.9000	91.82
Inception-v3	0.9969	0.9969	0.9968	99.69
ResNet-50	0.9969	0.9969	0.9968	99.69
VGG16	0.9642	0.9591	0.9600	95.91
Finetuned EfficientNet-B0	0.9969	0.9969	0.9969	99.69

Table 5. Performance comparison with other state-of-the-art models using the APV dataset.

Model	Precision	Recall	F1 Score	Accuracy (%)
EfficientNet-B0	0.7057	0.5028	0.4886	50.28
EfficientNet-B3	0.8822	0.7399	0.7630	73.99
Inception-v3	0.9925	0.9922	0.9922	99.22
ResNet-50	0.9928	0.9926	0.9926	99.26
VGG16	0.0002	0.0155	0.0005	1.50
Finetuned EfficientNet-B0	0.9979	0.9978	0.9978	99.78

Table 6. Performance comparison with other state-of-the-art models using the PV dataset.

Model	Number of parameters	Memory (MB)	#FLOPs (Millions)
EfficientNet-B0	4,099,530	15.64 MB	8.18 M
EfficientNet-B3	10,843,478	41.36 MB	21.71 M
Inception-v3	21,882,695	83.48 MB	43.73 M
ResNet-50	23,667,623	90.28 MB	47.25 M
VGG16	14,734,695	56.21 MB	29.43 M
Fine-tuned EfficientNet-B0	4,392,650	16.76 MB	8.85 M

Table 7. Comparison of memory with other recent state-of-the-art models.

optimal balance between model FLOPs, memory, and classification performance, rendering it highly suitable for deployment in environments where memory and computational resources are constrained, while still ensuring high accuracy in plant leaf classification tasks.

Discussion

Strategies for handling data imbalance

The study adopts a holistic training strategy to resolve data imbalance by integrating data augmentation, stratified data splitting, class weighting, and transfer learning approaches. These techniques collectively contribute to balanced representation, enhanced generalization, and improved classification performance. Importantly, the results observed in the study stem from the synergistic effect of these combined techniques rather than the isolated impact of individual techniques.

To rigorously validate this approach, a series of controlled ablation experiments were conducted on the APV dataset using the fine-tuned EfficientNet-B0 model. Each component of the hybrid strategy was individually removed within an otherwise consistent training pipeline.

The APV dataset was selected due to its moderate class imbalance and narrowly defined scope (apple leaf diseases), making it suitable for controlled experimentation. In this dataset, healthy leaves dominate, representing the majority class (51.88% of the dataset). In contrast, apple scab (19.81%), black rot (19.49%), and cedar apple rust (8.80%) represent minority classes. This distribution provides a sufficiently imbalanced scenario for evaluating the impact of data augmentation, stratified data splitting, and class weighting techniques. Conversely, the broader PV dataset exhibits higher class variability and more severe imbalance, limiting its suitability for isolating the impact of individual training strategies.

1. **Combined impact on data representation:** Data augmentation increases diversity within the training set, class weighting ensures proportional emphasis on minority classes, and stratification maintains consistent class distributions across training, validation, and test subsets. In order to increase model generalization, these techniques work in tandem to guarantee that minority classes are sufficiently represented and learned during training.

Table 8 (Page 19) compares the performance of the fine-tuned EfficientNet-B0 model with and without data augmentation on the APV dataset. A close analysis of these results reveals that data augmentation significantly enhanced classification performance, particularly for minority classes. For example, recall for apple scab improved from 0.76 to 1.00, and for cedar apple rust from 1.00 (with lower precision) to a more balanced 0.96, leading to F1 score increases from 0.86 and 0.85 to 0.99 and 0.98, respectively. Overall model accuracy increased from 93.40% to 99.69%, and the macro-averaged F1 score improved from 0.91 to 0.99, indicating better generalization and more balanced performance across all classes. No signs of overfitting or class bias were observed, as the healthy class retained consistently high performance in both settings. These results imply that data augmentation effectively improved class balance by synthetically increasing training diversity without introducing bias toward dominant class. When used alongside stratified data splitting and class weighting, the data augmentation strategy proved essential for achieving high accuracy, generalization, and fairness in multi-class plant disease classification.

It is important to emphasize here that varying data augmentation techniques substantially enhanced feature retention and classification consistency. The addition of transformations such as flips, rotations, zooms, and brightness adjustments, allowed the model to capture richer disease-specific features. As shown in Table 8, this improved generalization, particularly for minority classes. Furthermore, data augmentation fostered a more balanced and stable classification performance, mitigating dataset biases and enhancing model robustness.

Class / metric	With augmentation			Without augmentation		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Apple Scab	0.98	1.00	0.99	1.00	0.76	0.86
Black Rot	1.00	1.00	1.00	1.00	0.95	0.98
Cedar Apple Rust	1.00	0.96	0.98	0.74	1.00	0.85
Healthy	1.00	1.00	1.00	0.94	0.98	0.96
Macro Avg F1 Score	0.99			0.91		
Model-level Accuracy	99.69%			93.40%		

Table 8. Performance comparison of the fine-tuned EfficientNet-B0 model with and without data augmentation.

Class / metric	With stratified splitting			Without stratified splitting		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Apple Scab	0.98	1.00	0.99	0.47	0.98	0.63
Black Rot	1.00	1.00	1.00	0.94	1.00	0.97
Cedar Apple Rust	1.00	0.96	0.98	1.00	0.48	0.65
Healthy	1.00	1.00	1.00	0.99	0.70	0.82
Macro Avg F1 Score	0.99			0.77		
Model-level Accuracy	99.69%			78.93%		

Table 9. Performance comparison of the fine-tuned EfficientNet-B0 model with and without stratified splitting.

Class / metric	With class weighting			Without class weighting		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Apple Scab	0.98	1.00	0.99	1.00	0.60	0.75
Black Rot	1.00	1.00	1.00	1.00	0.65	0.78
Cedar Apple Rust	1.00	0.96	0.98	0.83	0.71	0.77
Healthy	1.00	1.00	1.00	0.76	1.00	0.87
Macro Avg F1 Score	0.99			0.79		
Model-level Accuracy	99.69%			82.70%		

Table 10. Performance comparison of the fine-tuned EfficientNet-B0 model with and without class weighting.

Table 9 (Page 19) presents the performance comparison of the fine-tuned EfficientNet-B0 model with and without stratified data splitting on the APV dataset. Removing stratified splitting led to a notable drop in overall test accuracy from 99.69% to 78.93%, and the macro-averaged F1 score declined from 0.99 to 0.77, indicating imbalanced classification performance. The cedar apple rust class, a minority category, exhibited severe degradation, with recall falling from 0.96 to 0.48 despite precision remaining at 1.00-highlighting a rise in false negatives. In comparison, the apple scab class showed high recall (0.98) but poor precision (0.47), suggesting over-prediction and class confusion caused by skewed distributions in non-stratified splits. These findings emphasize the significance of stratified splitting in maintaining class distribution across training, validation, and test sets. It mitigates bias toward majority classes, improves generalization, and ensures more consistent and reliable performance across all categories-particularly in real-world plant disease classification tasks involving imbalanced datasets.

Table 10 (Page 19) presents the performance comparison of the fine-tuned EfficientNet-B0 model with and without class weighting on the APV dataset. The findings clearly demonstrate that eliminating class weighting substantially degraded performance, with overall accuracy dropping from 99.69% to 82.70% and the macro-averaged F1 score decreasing from 0.99 to 0.79, indicating reduced generalization to underrepresented classes. Specifically, recall for minority classes such as apple scab and black rot fell sharply from 1.00 (with weighting) to 0.60 and 0.65, respectively. These findings underscore the critical role of class weighting in addressing data imbalance. By balancing gradient contributions across classes, it improved convergence speed, prevented overfitting to the dominant healthy class, and enhanced generalization. As a result, the model achieved stable and high-performance classification across both majority and minority classes. This analysis reinforces that class weighting is a pivotal strategy for learning balanced class representations in imbalanced datasets, contributing to faster convergence, improved fairness, and greater reliability in multi-class disease classification tasks.

2. **Leveraging transfer learning for robustness:** Transfer learning further complements these techniques by initializing the model with pre-trained weights from ImageNet, providing a strong foundation for feature extraction. Fine-tuning the entire architecture allows the model to adapt effectively to the imbalanced APV and PV datasets, mitigating the impact of limited and skewed training data.

To assess the importance of fine-tuning, we compared two training strategies. In the first approach, the pre-trained EfficientNet-B0 base/backbone model served as a frozen feature extractor, where only the customized head layer was trained. In the second approach-our proposed method-all layers were unfrozen and fine-tuned on the APV dataset to fully adapt the network to the target domain. As shown in Table 11 (Page 20), the frozen feature extractor configuration achieved only 23.27% accuracy, indicating that ImageNet features alone were insufficient to capture the subtle disease-specific patterns present in the APV dataset. In contrast, the fine-tuned model attained a test accuracy of 99.69%, with a dramatic reduction in test loss from 1.4722 to 0.0242. These results demonstrate that full-model fine-tuning was essential for effective domain adaptation, enabling the network to learn plant-specific disease features and achieve high generalization performance.

The combined application of these techniques under the holistic training strategy enabled the fine-tuned EfficientNet-B0 model to excel in classifying plant diseases across both specialized APV and generalized PV datasets. The integration of these techniques yielded balanced performance across all classes. For example, the model achieved consistently high F1-scores for both majority and minority categories in the APV and PV datasets (Tables 3 and 4).

Additionally, the confusion matrices (Figs. 8 and 9) demonstrated near-perfect true positive rates across all classes, including minority classes, highlighting the model's ability to avoid bias toward the dominant healthy class. The strong diagonal dominance and minimal off-diagonal errors in both confusion matrices validate that the applied holistic training strategy:

- Mitigated class imbalance by improving the model's capacity to detect minority classes,
- Avoided overfitting to majority classes by regularizing the loss gradient,
- Ensured representative validation and test distributions,
- And ultimately enabled domain-specific adaptation.

The observed balanced per-class performance and minimal false negatives for minority classes are direct consequences of this carefully designed training pipeline.

By employing these techniques holistically, the study achieves equitable performance across all classes without overfitting to majority categories or neglecting minority ones. The observed improvements, such as high F1-scores and robust classification accuracy, reflect the collective impact of data augmentation, class weighting, stratification, and transfer learning. This comprehensive approach ensures the model's reliability, robustness, and suitability for real-world agricultural applications, where addressing data imbalance is critical for actionable and equitable outcomes.

Optimizer selection and comparison

To assess optimizer effectiveness, we compared Adam, stochastic gradient descent (SGD) with momentum (0.9)⁸², root mean square propagation (RMSprop)⁸³, and AdaMax using three key evaluation metrics: training accuracy convergence, validation loss convergence (Fig. 10 (Page 21)), and final test accuracies (Table 12 (Page 21)). Training accuracy convergence curves in 10(a) reflected the learning speed and convergence stability of each optimizer, while validation loss convergence curves in Fig. 10(b) provided insights into generalization behavior and potential overfitting. The final test accuracy in Table 12 served as an unbiased measure of real-world performance. Collectively, these metrics offered a comprehensive understanding of each optimizer's training dynamics and generalization capability.

Adam demonstrated rapid gains in training accuracy within 8–10 epochs; however, its convergence performance was unsteady, and validation loss plateaued early, indicating signs of overfitting. This was confirmed by its poor generalization, yielding a final test accuracy of only 57.86%. RMSprop exhibited moderately stable validation performance and converged faster than SGD, achieving a test accuracy of 89.31%. SGD with momentum converged more slowly but steadily, with progressive improvements in validation loss, ultimately reaching a test accuracy of 98.11%.

Among all the optimizers tested, AdaMax delivered the most balanced and superior performance. It converged within 11–12 epochs, maintaining stable training accuracy and a consistent downward trend in validation loss. The model trained with AdaMax achieved a final test accuracy of 99.69%. This performance is attributed to AdaMax's use of the infinity norm, which handles sparse gradients effectively and complements our model architecture enhanced with dropout, GMP, and combined L1/L2 regularization. Based on its consistent and high-performing behavior across all metrics, AdaMax was selected as the optimizer for the proposed model.

Training strategy	Test accuracy (%)	Test loss
Frozen Feature Extractor (No Fine-tuning)	23.27	1.4722
Fine-Tuning Entire Model (Proposed fine-tuned EfficientNet-B0)	99.69	0.0242

Table 11. Impact of fine-tuning vs. feature extraction on APV dataset performance.

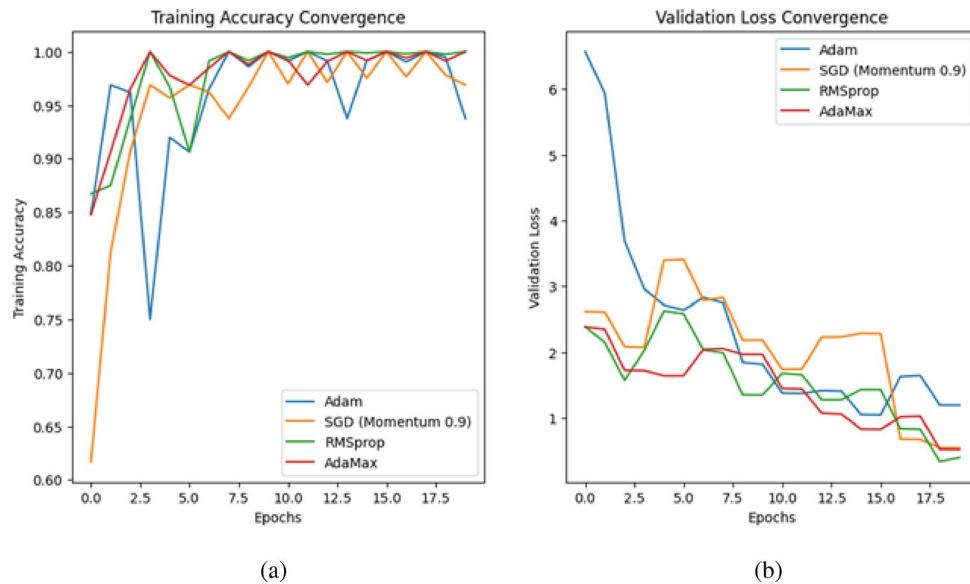


Fig. 10. Convergence curves for the fine-tuned EfficientNet-B0 model using the APV dataset: **(a)** training accuracy and **(b)** validation loss.

Optimizer	Convergence speed (Epochs)	Test accuracy (%)
Adam	8–10	57.86
RMSprop	10–12	89.31
SGD (Momentum 0.9)	15+	98.11
AdaMax	11–12	99.69

Table 12. Comparison of optimizer performance in terms of convergence behavior and final test accuracy.

Performance and computational efficiency

The fine-tuned EfficientNet-B0 model achieves significant performance improvements over other architectures with only marginal increases in memory usage and computational requirements, owing to several carefully implemented optimizations:

- Transfer learning:** Using pre-trained weights from ImageNet allowed the model to utilize robust feature representations, reducing the need for extensive training. Fine-tuning all layers allowed the model to adapt to the unique features of the APV and PV datasets, capturing critical patterns specific to plant leaf diseases and achieving state-of-the-art classification performance with minimal computational overhead.
- Compound scaling principle:** EfficientNet-B0's unique design, which balances network depth, width, and resolution, was preserved during fine-tuning. This approach minimized memory usage and FLOPs while maintaining high accuracy.
- Architectural enhancements:** Replacing the GAP layer with GMP improved the detection of localized disease patterns, such as lesions and spots, by emphasizing the most salient features in the input and avoiding over-smoothing. Dropout layers further mitigated overfitting without adding computational complexity.
- Advanced regularization:** Techniques like L1 and L2 regularization were employed to prevent overfitting by penalizing large weights, thereby promoting generalization. This approach improved the model's accuracy on unseen data while maintaining a low memory footprint.
- Handling class imbalance:** Strategies such as data augmentation, stratification, and class weighting were applied to enhance generalizability and reduce unnecessary computations during training.

Collectively, these optimizations enabled the fine-tuned EfficientNet-B0 model to strike an optimal balance between accuracy, memory efficiency, and computational demands. As a result, the fine-tuned EfficientNet-B0 model emerges as a practical and scalable solution for real-world applications in smart agriculture, where resource constraints are a significant challenge.

Capabilities and limitations of the fine-tuned EfficientNet-B0 model

A fine-tuned EfficientNet-B0 model is specifically designed for classifying apple leaf diseases. The model achieves remarkably high accuracy while still being computationally and memory efficient, making it suitable for deployment on devices with restricted hardware capabilities. It is important to emphasize that there exist

alternative CNN algorithms that provide greater accuracy rates but need more processing complexity and memory resources. Striking a balance between the memory requirements, computing efficiency, and accuracy continues to be a persistent challenge in DL-based plant leaf disease classification. In this context, the fine-tuned model delivers astonishingly high accuracies with low memory and FLOPs requirements.

It is important to emphasize that the proposed fine-tuned model trained on datasets such as the PV and APV datasets may face challenges when applied to novel environments, different plant species, or other disease presentations. This limitation arises from the controlled conditions and limited diversity inherent in these datasets. Hence, it is crucial to augment these datasets with samples from many sources and to consider environmental aspects while modeling to improve the ability to generalize. It is also imperative to tackle the issues related to imbalanced datasets for plant diseases. It is crucial to overcome these obstacles to strengthen the resilience and practical usefulness of the proposed fine-tuned model in real-life situations. Obtaining larger and more varied datasets will be crucial in accomplishing this goal.

Additionally, the proposed model's effectiveness may be limited when presented with images with intricate backgrounds, due to its primary training on uncomplicated leaf backdrops. Future research endeavors could also investigate the use of ensemble approaches to enhance classification accuracy, especially in challenges involving several classes.

Applications in smart agriculture systems

The fine-tuned EfficientNet-B0 model, with its compact design (16.76 MB memory usage and 8.85 million FLOPs), allows rapid inference speed and real-time deployment on edge computing devices, mobile phones, and agricultural drones. Its computational efficiency ensures autonomous field operation without reliance on cloud infrastructure, addressing the practical needs of real-world agricultural deployments. Also, this high-accuracy and lightweight model presents transformative opportunities for precision agriculture by integrating artificial intelligence (AI), machine vision, and Internet of Things (IoT)-driven technologies into smart farming ecosystems. Furthermore, its scalability supports both large-scale and smallholder agricultural operations in diverse environmental conditions.

By embedding the model into edge devices such as drones, robotic systems, and smart pesticide sprayers^{79–81}, it becomes possible to conduct real-time diagnostics of plant health. Such applications are essential for mitigating crop losses caused by delayed interventions. The proposed model's ability to classify leaf diseases with near-perfect accuracy facilitates targeted pesticide application, reducing chemical use, operational costs, and environmental footprint. For instance, the system can identify diseases like powdery mildew in vineyards at early stages and guide autonomous sprayers to treat only affected areas. Similarly, in cereal crop farming, it enables timely interventions against rust diseases, preserving yield quality.

The integration of the proposed model into automated variable-rate pesticide spraying systems^{80,81} transforms resource utilization by tailoring agrochemical application to the severity and location of infections. These systems conserve chemicals, prevent runoff, and align with sustainable agricultural practices. The model's low memory and computational requirements make it deployable even in resource-constrained environments, empowering smallholder farmers to access advanced technology at a lower cost.

Scalability across diverse agricultural operations is a key feature of the proposed model. The model is equally effective in small-scale, high-value crop settings as in extensive industrial farming. Its adaptability to various crop types underscores its utility in addressing the heterogeneity of agricultural systems. It is deployable in different climatic and environmental conditions when fine-tuned with localized datasets.

Additionally, the model supports real-time analysis, enabling dynamic adjustments during field operations. For example, mounted on a mobile device or drone, it can continuously scan for symptoms of diseases like apple scab or black rot, triggering precise responses that enhance operational efficiency. This capability addresses the critical need for speed and accuracy in modern agriculture, where delays in intervention can lead to significant yield losses.

Future work should include field trials to validate its theoretical advantages and optimize its integration for real-world conditions, ensuring broader applicability and practical impact.

Conclusion and future work

In this study, a fine-tuned EfficientNet-B0 model was proposed, targeting low memory usage and FLOPs, while also attaining high accuracy in apple leaf disease classification performance. Primarily, an EfficientNet-B0 model with pre-trained weights was employed as the base model. The head layer of the model was customized to meet our specific performance requirements. A key architectural improvement in the head layer was GMP, which prioritized localized feature activations crucial for disease identification. This head layer customization step was then followed by comprehensive fine-tuning of the entire model. To address class imbalance and improve generalization, this study adopted a holistic training strategy that combined data augmentation, stratified data splitting, and class weighting in conjunction with transfer learning. The model achieved exceptional precision, recall, and F1 scores, and exhibited outstanding test accuracies of 99.69% and 99.78% in successfully classifying different plant diseases using APV and PV datasets, respectively. The fine-tuned model outperformed well-known architectures like EfficientNet-B0, EfficientNet-B3, and VGG16 in terms of accuracy on the APV dataset. The fine-tuned model also outperformed EfficientNet-B0, EfficientNet-B3, Inception-v3, ResNet-50, and VGG16 in terms of accuracy on the PV dataset. The EfficientNet-B0 and fine-tuned EfficientNet-B0 emerged as the most memory and computationally efficient choices among the comparative models. However, the fine-tuned model demonstrated its edge over the EfficientNet-B0 with an 11% improvement in accuracy on the APV dataset and a remarkable 49.5% improvement in accuracy on the PV dataset. Remarkably, this performance improvement came at an approximate 7-8% increase in both memory use and FLOPs. The fine-tuned EfficientNet-B0 model thus emerged as a robust model, achieving exceptional accuracy while maintaining optimized memory and

FLOPs usage, making it highly suitable for accurate classification of plant leaf diseases with high inference speed in resource-constrained environments.

The results of our ablation study confirmed that the combined application of data augmentation, class weighting, and stratified data splitting, including transfer-learning, was instrumental in achieving robust and balanced classification performance across both majority and minority classes. While each technique individually contributed to model improvement, it was their integration that enabled the fine-tuned EfficientNet-B0 model to attain high generalization, minimize class-wise bias, and maintain consistent performance across diverse and imbalanced datasets. This holistic strategy proved essential for developing a scalable and fair plant disease classification framework suited for real-world agricultural environments.

Overall, this study suggests that combining fine-tuned CNN techniques with transfer learning, a holistic training strategy, and architectural optimizations can significantly enhance the accuracy and generalization of CNN models for classifying plant leaf diseases. The fine-tuned approach also allows for more efficient implementation in situations with limited resources.

While the model's lightweight design makes it suitable for deployment on edge devices such as mobile phones and drones, certain limitations were noted. These include the need for diverse datasets to improve generalization across novel environments and the challenges posed by complex image backgrounds. To address these limitations, the automatic diagnostic capability of the fine-tuned model will be further improved by expanding the datasets and enhancing the model. Future research will also prioritize the integration of the fine-tuned model with machine vision cameras for the research and development (R&D) work of AI-driven automated variable rate pesticide sprayers. The integration of machine vision and DL technology in smart sprayers will enable precise and efficient pesticide application, led by real-time, data-driven insights. This approach will reduce chemical waste, enhance resource efficiency, and promote sustainable farming practices.

Moreover, our future efforts will prioritize the incorporation of multi-sensor data into our model to achieve a thorough comprehension of plant health and greenhouse conditions. This will enable the development of more robust and flexible systems for classification of plant diseases in smart agriculture.

We recognize that real-world agricultural datasets often suffer from complex and evolving imbalance. In future work, we intend to explore advanced strategies, such as class-specific decision threshold optimization⁸⁴, focal loss⁸⁵, generative augmentation⁸⁶, and contrastive learning⁸⁷, to further enhance minority class recognition, reduce bias, and maximize generalization in plant disease classification tasks. These approaches will build upon our current hybrid strategy and help extend model applicability to large-scale, field-level deployment where class imbalance is more pronounced and data quality may vary.

While the model demonstrated strong generalization on both the APV and PV datasets, future work will also focus on enhancing adaptability to unseen plant diseases and diverse field conditions. This includes integrating domain adaptation techniques⁸⁸, few-shot meta-learning approaches such as model-agnostic meta-learning (MAML)⁸⁹, and sensor fusion strategies combining RGB with multi-spectral or thermal imaging. In addition, expanding real-world validation through collaborative field studies will help improve robustness and ensure practical scalability.

Data availability

The PV dataset used for this research work is taken from: <https://data.mendeley.com/datasets/tywbtsjrjv/1>

Received: 28 September 2024; Accepted: 27 May 2025

Published online: 16 July 2025

References

- Food and agriculture organization of the united nations. In *FAOSTAT Database*. <https://www.fao.org/faostat/en/home> (2024).
- Bansal, P., Kumar, R. & Kumar, S. Disease detection in apple leaves using deep convolutional neural network. *Agriculture* **11**, 617. <https://doi.org/10.3390/agriculture11070617> (2021).
- Khan, A., Nawaz, U., Ulhaq, A. & Robinson, R. W. Real-time plant health assessment via implementing cloud-based scalable transfer learning on AWS DeepLens. *PLoS ONE* **15**, e0243243. <https://doi.org/10.1371/journal.pone.0243243> (2020).
- Goldberg, N. P. *Apple Disease Control: Guide H-317*. New Mexico State University and U.S. Department of Agriculture. https://pubs.nmsu.edu/_h/H317.pdf (2000).
- Sai, A. M. & Patil, N. Comparative analysis of machine learning algorithms for disease detection in apple leaves. In *Proceedings of the 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Shivamogga, India* 239–244 (2022). <https://doi.org/10.1109/DISCOVER55800.2022.9974840>.
- Anam, S. Segmentation of leaf spots disease in apple plants using particle swarm optimization and K-means algorithm. *J. Phys.: Conf. Ser.* **1562**, 012011. <https://doi.org/10.1088/1742-6596/1562/1/012011> (2020).
- Khan, M. A. et al. An optimized method for segmentation and classification of apple diseases based on strong correlation and genetic algorithm based feature selection. *IEEE Access* **7**, 46261–46277. <https://doi.org/10.1109/ACCESS.2019.2908040> (2019).
- Bracino, A. A., Concepcion, R. S., Bedruz, R. A. R., Dadios, E. P. & Vicerra, R. R. P. Development of a hybrid machine learning model for apple (*Malus domestica*) health detection and disease classification. In *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)* 1–6 (IEEE, 2020). <https://doi.org/10.1109/HNICEM51456.2020.9400139>.
- Chakraborty, S., Paul, S. & Rahat-uz-Zaman, M. Prediction of apple leaf diseases using multiclass support vector machine. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* 1–6 (IEEE, 2021). <https://doi.org/10.1109/icrest51555.2021.9331132>.
- Zhong, Y. & Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **168**, 105146. <https://doi.org/10.1016/j.compag.2019.105146> (2020).
- Yu, H.-J. & Son, C.-H. Leaf spot attention network for apple leaf disease identification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 229–237 (IEEE, 2020). <https://doi.org/10.1109/CVPRW50498.2020.00034>.
- Rehman, M. Z. U. et al. Recognizing apple leaf diseases using a novel parallel real-time processing framework based on Mask RCNN and transfer learning: an application for smart agriculture. *IET Image Process.* **15**, 2157–2168 (2021).

13. Luo, Y. et al. Apple leaf disease recognition and sub-class categorization based on improved multi-scale feature fusion network. *IEEE Access* **9**, 95517–95527. <https://doi.org/10.1109/ACCESS.2021.3094802> (2021).
14. Tian, Y., Li, E., Liang, Z., Tan, M. & He, X. Diagnosis of typical apple diseases: a deep learning method based on multi-scale dense classification network. *Front. Plant Sci.* **12**, 698474. <https://doi.org/10.3389/fpls.2021.698474> (2021).
15. Li, L., Zhang, S. & Wang, B. Apple leaf disease identification with a small and imbalanced dataset based on lightweight convolutional networks. *Sensors* **22**, 173. <https://doi.org/10.3390/s22010173> (2021).
16. Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K. & Dollár, P. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10425–10433 (IEEE, 2020). <https://doi.org/10.1109/CVPR42600.2020.01044>.
17. Xie, Q., Luong, M.-T., Hovy, E. & Le, Q. V. Self-training with noisy student improves ImageNet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10687–10698 (IEEE, 2020). <https://doi.org/10.1109/CVPR42600.2020.01070>.
18. Yang, Q., Duan, S. & Wang, L. Efficient identification of apple leaf diseases in the wild using convolutional neural networks. *Agronomy* **12**, 2784 (2022) <https://www.mdpi.com/2073-4395/12/11/2784>.
19. Alqahtani, M. et al. Sailfish optimizer with EfficientNet model for apple leaf disease detection. *Comput. Mater. Continua* **74**, 217–233. <https://doi.org/10.32604/cmc.2023.025280> (2023).
20. Zhao, G. & Huang, X. Apple leaf disease recognition based on improved convolutional neural network with an attention mechanism. In *Proceedings of the 2023 5th International Conference on Natural Language Processing (ICNLP)* 98–102 (2023). <https://doi.org/10.1109/ICNLP58431.2023.00024>.
21. Liu, Z. & Li, X. An improved YOLOv5-based apple leaf disease detection method. *Sci. Rep.* **14**, 17508. <https://doi.org/10.1038/s41598-024-67924-8> (2024).
22. Di, J. & Li, Q. A method of detecting apple leaf diseases based on improved convolutional neural network. *PLOS ONE* **17**, e0262629. <https://doi.org/10.1371/journal.pone.0262629> (2022).
23. Gupta, S. & Tripathi, A. K. Fruit and vegetable disease detection and classification: recent trends, challenges, and future opportunities. *Eng. Appl. Artif. Intell.* **133**, 108260. <https://doi.org/10.1016/j.engappai.2024.108260> (2024).
24. Boulent, J., Foucher, S., Théau, J. & St-Charles, P.-L. Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* **10**, 941. <https://doi.org/10.3389/fpls.2019.00941> (2019).
25. Fraiwan, M., Faouri, E. & Khasawneh, N. On using deep artificial intelligence to automatically detect apple diseases from leaf images. *Sustainability* **14**, 10322. <https://doi.org/10.3390/su141610322> (2022).
26. Tan, M. & Le, Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2020).
27. Li, B., Liu, B., Li, S. & Liu, H. An improved EfficientNet for rice germ integrity classification and recognition. *Agriculture* **12**, 863. <https://doi.org/10.3390/agriculture12060863> (2022).
28. Arun, Y. & Viknesh, G. S. Leaf classification for plant recognition using EfficientNet architecture. In *2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAEEC)* 1–5 (IEEE, 2022). <https://doi.org/10.1109/ICAEEC54045.2022.9716637>.
29. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J. & Greenspan, H. Synthetic Data Augmentation using GAN for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* 289–293 (IEEE, 2018). <https://doi.org/10.1109/ISBI.2018.8363576>.
30. Mumuni, A. & Mumuni, F. Data augmentation: a comprehensive survey of modern approaches. *Array* **16**, 100258. <https://doi.org/10.1016/j.array.2022.100258> (2022).
31. Yang, B. et al. A novel plant type, leaf disease and severity identification framework using CNN and transformer with multi-label method. *Sci. Rep.* **14**, 11664. <https://doi.org/10.1038/s41598-024-62452-x> (2024).
32. Sadayandi, J., Arumugam, P., Sangaiah, A. K. & Zhang, C. Stratified sampling-based deep learning approach to increase prediction accuracy of unbalanced dataset. *Electronics* **12**, 4423. <https://doi.org/10.3390/electronics12214423> (2023).
33. Scikit-learn developers. Sklearn.utils.class_weight.compute_class_weight. https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html (2024).
34. Pandian, J. A. & Gopal, G. Data for: identification of plant leaf diseases using a 9-layer deep convolutional neural network. *Mendeley Data* **2019**, V1. <https://doi.org/10.17632/tywbtjsrjv.1> (2019).
35. Andrew, J., Eunice, J., Popescu, D. E., Chowdary, M. K. & Hemanth, J. Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy* **12**, 2395. <https://doi.org/10.3390/agronomy12102395> (2022).
36. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2015).
37. Kavalas, S. A. P. N. & Pothuraju, R. Detection of grape leaf disease using transfer learning methods: VGG16 & VGG19. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)* 1205–1208 (IEEE, 2022). <https://doi.org/10.1109/ICCMC53470.2022.9753773>.
38. Gu, J., Yu, P., Lu, X. & Ding, W. Leaf species recognition based on VGG16 networks and transfer learning. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* 2189–2193 (IEEE, 2021). <https://doi.org/10.1109/IAEAC50856.2021.9390789>.
39. Swasono, D. I., Tjandrasa, H. & Fathicah, C. Classification of tobacco leaf pests using VGG16 transfer learning. In *2019 International Conference on Information and Communication Technology and Systems (ICTS)* 176–181 (IEEE, 2019). <https://doi.org/10.1109/ICTS500946>.
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2818–2826 (IEEE, 2016). <https://doi.org/10.1109/CVPR.2016.308>.
41. Liu, Y. & Zheng, Y. F. One-against-all multi-class SVM classification using reliability measures. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, vol. 2* 849–854 (IEEE, 2005). <https://doi.org/10.1109/IJCNN.2005.1555963>.
42. Yadav, R., Rana, Y. K. & Nagpal, S. Plant leaf disease detection and classification using particle swarm optimization. In *Lecture Notes in Computer Science* 294–306 (Springer, 2019). https://doi.org/10.1007/978-3-030-19945-6_21.
43. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90. <https://doi.org/10.1145/3065386> (2017).
44. Jiang, P., Chen, Y., Liu, B., He, D. & Liang, C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* **7**, 59069–59080. <https://doi.org/10.1109/ACCESS.2019.2914929> (2019).
45. Howard, A. G. et al. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
46. Bi, C. et al. MobileNet based apple leaf diseases identification. *Mob. Netw. Appl.* **27**, 172–180. <https://doi.org/10.1007/s11036-020-01640-1> (2022).
47. Yu, H.-J. & Son, C.-H. Apple leaf disease identification through region-of-interest-aware deep convolutional neural network. *arXiv preprint arXiv:1903.10356* (2019).
48. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385* (2015).
49. Huang, G., Liu, Z., Van der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2261–2269 (IEEE, 2017). <https://doi.org/10.1109/CVPR.2017.243>.
50. Xu, X. et al. Plant leaf disease identification by parameter-efficient transformer with adapter. *Eng. Appl. Artif. Intell.* **138**, 109466. <https://doi.org/10.1016/j.engappai.2024.109466> (2024).

51. Sharma, V., Tripathi, A. K. & Mittal, H. CLDA-Net: a novel citrus leaf disease attention network for early identification of leaf diseases. In *2023 15th International Conference on Computer and Automation Engineering (ICCAE)* 178–182 (IEEE, 2023). <https://doi.org/10.1109/ICCAE56788.2023.10111244>.
52. Ahmad, M., Abdullah, M., Moon, H. & Han, D. Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning. *IEEE Access* **9**, 140565–140580. <https://doi.org/10.1109/ACCESS.2021.3119655> (2021).
53. Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z. & Jasińska, E. Identification of plant-leaf diseases using CNN and transfer-learning approach. *Electronics* **10**, 1388. <https://doi.org/10.3390/electronics10121388> (2021).
54. ImageNet. ImageNet: a large-scale hierarchical image database. <https://image-net.org/> (2024).
55. Amara, J., Bouaziz, B. & Algergawy, A. A deep learning-based approach for banana leaf diseases classification. In *Datenbanksysteme für Business, Technologie und Web (BTW)* (2017). <https://api.semanticscholar.org/CorpusID:38788670>.
56. Zhang, K., Wu, Q., Liu, A. & Meng, X. Can deep learning identify tomato leaf disease? *Adv. Multimedia* **2018**, 6710865. <https://doi.org/10.1155/2018/6710865> (2018).
57. Cruz, A. C., Luvisi, A., De Bellis, L. & Ampatzidis, Y. Vision-based plant disease detection system using transfer and deep learning. In *2017 ASABE Annual International Meeting*, (American Society of Agricultural and Biological Engineers, 2017). <https://doi.org/10.13031/aim.201700241>.
58. Brahim, M., Boukhalfa, K. & Moussaoui, A. Deep learning for Tomato diseases: classification and symptoms visualization. *Appl. Artif. Intell.* **31**, 299–315. <https://doi.org/10.1080/08839514.2017.1315516> (2017).
59. Sibiya, M. & Sumbwanyambe, M. A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. *AgriEngineering* **1**, 119–131. <https://doi.org/10.3390/agriengineering1010009> (2019).
60. Geiger, A., Lauer, M., Wojek, C., Stiller, C. & Urtasun, R. 3D traffic scene understanding from movable platforms. *IEEE Trans Pattern Anal. Mach. Intell.* **36**, 1012–1025. <https://doi.org/10.1109/TPAMI.2013.185> (2014).
61. Iman, M., Arabnia, H. R. & Rasheed, K. A review of deep transfer learning and recent advancements. *Technol. (Basel)* **11**, 40. <https://doi.org/10.3390/technologies11020040> (2023).
62. Shrestha, G., Deepshikha & Das, M. Plant disease detection using CNN. In *2020 IEEE Applied Signal Processing Conference (ASP CON)* 109–113 (IEEE, 2020). <https://doi.org/10.1109/ASPCON49795.2020.9276722>.
63. Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z. & Jasińska, E. Identification of plant-Leaf diseases using CNN and transfer-learning approach. *Electronics* **10**, 1388. <https://doi.org/10.3390/electronics10121388> (2021).
64. Renuka, J. G., Likhitha, G., Modala, V. K. & Reddy, D. M. Plant disease detection using fine-tuned ResNet architecture. In *Lecture Notes in Networks and Systems* 527–541 (Springer, 2023). <https://doi.org/10.1007/978-981-99-3010-4-44>.
65. Islam, M. M. et al. A deep learning model for cotton disease prediction using fine-tuning with smart web application in agriculture. *Intell. Syst. Appl.* **20**, 200278. <https://doi.org/10.1016/j.iswa.2023.200278> (2023).
66. Zafar, A. et al. A comparison of pooling methods for convolutional neural networks. *Appl. Sci.* **12**, 8643. <https://doi.org/10.3390/app12178643> (2022).
67. Boureau, Y.-L., Ponce, J. & LeCun, Y. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML)* 111–118 (Omnipress, 2010).
68. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* 448–456 (JMLR.org, 2015).
69. Too, E. C., Li, Y., Njuki, S. & Liu, Y. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **161**, 272–279. <https://doi.org/10.1016/j.compag.2018.03.032> (2019).
70. Afzaal, H. et al. Detection of a potato disease (Early blight) using artificial intelligence. *Remote Sens.* **13**, 411. <https://doi.org/10.3390/rs13030411> (2021).
71. Yang, M., Lim, M. K., Qu, Y., Li, X. & Ni, D. Deep neural networks with L1 and L2 regularization for high dimensional corporate credit risk prediction. *Expert Syst. Appl.* **213**, 118873. <https://doi.org/10.1016/j.eswa.2022.118873> (2023).
72. Larsson, G., Maire, M. & Shakhnarovich, G. FractalNet: ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648* (2017).
73. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
74. Alabassy, B., Safar, M. & El-Kharashi, M. W. A high-accuracy implementation for softmax layer in deep neural networks. In *Proc. 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)* 1–6 (IEEE, 2020). <https://doi.org/10.1109/DTIS48698.2020.9081313>.
75. Azzawi, H., Hou, J., Xiang, Y. & Alanni, R. Lung cancer prediction from microarray data by gene expression programming. *IET Syst. Biol.* **10**, 168–178. <https://doi.org/10.1049/iet-syb.2015.0082> (2016).
76. Saleem, M. H., Potgieter, J. & Arif, K. M. Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers. *Plants* **9**, 1319. <https://doi.org/10.3390/plants9101319> (2020).
77. Llugsi, R., Yacoubi, S. E., Fontaine, A. & Lupera, P. Comparison between Adam, AdaMax and AdamW optimizers to implement a weather forecast based on neural networks for the Andean city of Quito. In *Proc. IEEE Ecuador Tech. Chapters Meet. (ETCM)* 1–6 (IEEE, 2021). <https://doi.org/10.1109/ETCM53643.2021.9590681>.
78. Sanida, M. V., Sanida, T., Sideris, A. & Dasgupta, M. An efficient hybrid CNN classification model for tomato crop disease. *Technology* **11**, 10. <https://doi.org/10.3390/technologies11010010> (2023).
79. Zhang, Z., Wang, X., Lai, Q. & Zhang, Z. Review of variable-rate sprayer applications based on real-time sensor technologies. In *Precision Agriculture Technologies for Food Security and Sustainability* (IntechOpen, 2018). <https://doi.org/10.5772/intechopen.73622>.
80. Zaman, Q. et al. Development of prototype automated variable rate sprayer for real-time spot-application of agrochemicals in wild berry fields. *Comput. Electron. Agric.* **76**, 175–182. <https://doi.org/10.1016/j.compag.2011.01.014> (2011).
81. Hussain, N. et al. Design and development of a smart variable rate sprayer using deep learning. *Remote Sens.* **12**, 4091. <https://doi.org/10.3390/rs12244091> (2020).
82. Jin, R. & He, X. Convergence of momentum-based stochastic gradient descent. In *Proc. 16th Int. Conf. Control Autom. (ICCA)* 779–784 (IEEE, 2020). <https://doi.org/10.1109/ICCA51439.2020.9264458>.
83. Xu, D., Zhang, S., Zhang, H. & Mandic, D. P. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Netw.* **139**, 17–23. <https://doi.org/10.1016/j.neunet.2021.02.011> (2021).
84. Zou, Q., Xie, S., Lin, Z., Wu, M. & Ju, Y. Finding the best classification threshold in imbalanced classification. *Big Data Res.* **5**, 2–8. <https://doi.org/10.1016/j.bdr.2015.12.001> (2016).
85. Liu, C., E. X. & Wang, S. Lightweight model for apple leaf disease recognition based on Haar wavelet downsampling and focal loss function. In *Proc. 2024 Int. Conf. Image Process. Comput. Vis. Mach. Learn. (ICICML)* 901–906 (IEEE, 2024). <https://doi.org/10.1109/ICICML63543.2024.10957849>.
86. Zhang, H., Zhang, Q., Yan, W. & Gu, J. A generative data augmentation enhanced plant disease classification combined diffusion model. In *Proc. 4th Int. Conf. Electron. Inf. Eng. Comput. Commun. (EIECC)* 874–878 (IEEE, 2024). <https://doi.org/10.1109/EIEC64539.2024.10929258>.
87. Bai, Z., Xu, H., Ding, Q. & Zhang, X. Feature contrastive transfer learning for few-shot long-tail sonar image classification. *IEEE Commun. Lett.* **29**, 562–566 (2025).

88. Xu, Q., Yuan, X. & Ouyang, C. Class-aware domain adaptation for semantic segmentation of remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–17. <https://doi.org/10.1109/TGRS.2020.3031926> (2022).
89. Shao, Y., Wu, W., You, X., Gao, C. & Sang, N. Improving the generalization of MAML in few-shot classification via bi-level constraint. *IEEE Trans. Circ. Syst. Video Technol.* **33**, 3284–3295. <https://doi.org/10.1109/TCSVT.2022.3232717> (2023).

Acknowledgements

The authors would like to thank the Qatar Research, Development and Innovation (QRDI) Council for funding this research under grant number MME03-1121-210025.

Author contributions

H.A. and N.S. conceptualized the study. H.A. and N.S. developed the methodology and contributed to software development. H.A., R.B., and R.Y. contributed to validation and performed formal analysis. H.A. and N.S. carried out the investigation and curated the data. R.B. and A.A.F. provided resources. H.A. and N.S. wrote the original draft, with R.B. and A.A.F. contributing to review and editing. H.A., and R.Y. contributed to visualization. R.B. and A.A.F. supervised the project. H.A. and R.B. administered the project, and R.B. acquired funding. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Qatar Research, Development and Innovation (QRDI) Council grant number MME03-1121-210025.

Declarations

The data presented in Table 1 is sourced from²⁶. The original source is duly acknowledged in the Table 1 caption.

Competing interests

The authors declare no conflicts of interest.

Consent to participate

All Authors agree to participate in the published version of the manuscript.

Additional information

Correspondence and requests for materials should be addressed to H.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025