

A Lightweight Convolutional Neural Network Framework for Automated Visual Classification of Maize Leaf Diseases

Kayombo Sakachiva

kayombosakachiva84@gmail.com

ZCAS University

Jackson Phiri

University of Zambia

Mayumbo Nyirenda

University of Zambia

Research Article

Keywords: maize leaf disease, convolutional neural network, MobileNetV2, MobileNetV3-Small, EfficientNet-B0, transfer learning, lightweight CNN, ONNX, mobile deployment, image classification

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11164174/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

A Lightweight Convolutional Neural Network Framework for Automated Visual Classification of Maize Leaf Diseases

Kayombo Sakachiva: School of Computing, Technology and Applied Sciences, ZCAS University, Lusaka, Zambia. Email: kayombosakachiva84@gmail.com. Jackson Phiri: Department of Computer Science, University of Zambia, Lusaka, Zambia Email: jackson.phiri@cs.unza.zm. Mayumbo Nyirenda: Department of Computer Science, University of Zambia, Lusaka, Zambia. Email: mayumbo@gmail.com

“Corresponding author: Kayombo Sakachiva”

Abstract

Maize leaf disease may have detrimental effects on the health and productivity of crops, while traditional visual diagnosis may face limitations owing to the availability of agricultural experts and the conditions of disease identification. Artificial neural networks provide an opportunity for automated visual learning from maize leaf images, enabling disease-related visual patterns to be learned and classified directly from image data. This research presents a controlled comparison between three lightweight CNNs (MobileNetV2, MobileNetV3-Small, and EfficientNet-B0) in classifying four types of maize leaf diseases. An initial dataset comprising 4,188 images of Blight, Common Rust, Gray Leaf Spot, and Healthy leaves was used in this research. After data quality assessment and removal of duplicates, eight duplicated images were removed from the dataset, leaving 4,180 unique images for model building. The dataset was split randomly using seed number 42 into training, validation, and independent test sets consisting of 2,927, 628, and 625 images, respectively.

Classification performance was measured using accuracy, macro-averaged precision, recall, F1-score, confusion matrix, and ROC-AUC. Computational and deployability properties were assessed using parameter count, ONNX model size, training time, desktop inference latency, and Android inference latency. MobileNetV2 obtained an accuracy of 96.80% and a macro F1-score of 96.07%, MobileNetV3-Small obtained an accuracy of 96.00% and a macro F1-score of 95.04%, and EfficientNet-B0 obtained an accuracy of 96.96% and a macro F1-score of 96.34%. Overall ROC-AUC scores were 99.51%, 99.72%, and 99.82%, respectively.

Keywords: maize leaf disease, convolutional neural network, MobileNetV2, MobileNetV3-Small, EfficientNet-B0, transfer learning, lightweight CNN, ONNX, mobile deployment, image classification.

SECTION I

A. Introduction

Corn or (Maize) *Zea mays* L. is one of the staple food crops cultivated worldwide and also one of the food security staples in sub-Saharan Africa. Maize production is susceptible to a number of diseases including blight, common rust, and gray leaf spot which have the potential to affect the crop yield and quality. Timely identification of diseases is therefore an important step in order to enable intervention in time and minimize crop losses. Disease identification through visual inspection usually involves farmer observation, inspection by agricultural extension officers and plant pathology specialists. Although expert observation can help identify diseases, this process may take time, be subject to human error, and may not always be available, especially when agricultural specialists are unavailable [1], [2], [3].

Artificial neural networks have increasingly enabled automated visual analysis in agriculture, particularly through deep-learning and computer-vision techniques for the identification and diagnosis of plant diseases from images [4].

The emergence of deep learning and computer vision methods in recent advancements in artificial intelligence (AI) has made automation in disease detection based on visual analysis possible. CNNs, in particular, play a very important role in the identification of plant diseases through images since these networks allow for automatic feature extraction directly from images without manual engineering of the features. Studies on plant disease recognition have highlighted the rise of CNN-based methods in image classification, disease recognition, and other tasks in agricultural computer vision [1], [3].

The use of deep learning in maize disease identification is no exception and has grown in recent years as well. Scientists have explored CNNs, transfer learning, residual nets, EfficientNet structures, vision transformers, and various other techniques using deep learning in order to diagnose diseases on maize leaves. This shows the applicability of deep learning in maize disease diagnosis in Africa's agriculture sector [5].

Despite all these advancements, having high classification accuracy is not enough to develop a functional agricultural visual-learning system. Many deep neural networks have huge amounts of parameters and consume lots of computational power, memory, and time. These factors can make

the application of these models on low-performance devices like smartphones, embedded computers, edge-AI devices, and other agricultural field technologies difficult. It is especially the case for applications for agriculture in developing countries where there might not always be a reliable high-performance computing infrastructure and permanent Internet connection [6].

Efforts have also been made to develop unified CNN-based frameworks capable of identifying diseases across multiple crops, including maize, rice, and wheat [7]. While both large and lightweight convolutional models have been applied to maize leaf disease recognition, the growing need for computational efficiency has also encouraged the development of hybrid lightweight architectures [8].

This study presents a controlled comparison of MobileNetV2, MobileNetV3-Small, and EfficientNet-B0 using the same maize leaf disease dataset, data split, preprocessing, training, and evaluation protocol. The study further validates the trained models through ONNX export and inference consistency testing, followed by real-device Android benchmarking to assess their practical deployment performance

The research does not introduce a new CNN architecture. Instead, its contribution lies in the controlled evaluation and comparison of established lightweight architectures under a common experimental protocol, together with an assessment of their portability to an offline mobile inference environment.

B. Background of the Study

Plant diseases pose a critical threat to agriculture because disease infections may adversely affect the growth, yield, and quality of plants. The early detection of diseases is especially crucial because late detection of a disease might allow the disease to spread before any necessary action is taken to control it. Traditional disease detection has relied on visual inspection of symptoms of disease by either farmers or agriculturalists. Visual inspection of plant disease may be subject to biases arising from human experience, environmental conditions, disease similarities, and availability of experts [2], [3].

There have been advancements in the use of cameras and smartphones, presenting an opportunity to employ artificial intelligence techniques using images for agricultural diagnosis. A farmer could take an image of a diseased plant leaf through his smartphone, and then an automatic learning

technique based on computer vision could analyze the image and predict the disease class. Convolutional Neural Networks are well suited to this task since they can detect the spatial pattern such as colors and lesions of diseases.

The advancements in deep learning have seen the rise of many research works in the application of CNNs for disease recognition in plants. Deep learning has become an important approach in plant disease detection because it can automatically extract relevant visual features such as color patterns, lesions, and texture from leaf images [3], [9]. Convolutional neural networks, in particular, have shown strong potential for image-based agricultural diagnosis.

The problem that arises due to complexity is related to the fact that in order to improve accuracy, the depth, width, or complexity of the neural network must be increased. While this increases representational power, it also increases computation needs.

The advancements are particularly important in the agricultural field. The disease detection algorithm that is meant to run on a computer will require more computing power than an algorithm meant to run on a mobile phone or edge device. Therefore, it is critical to select the appropriate algorithm based on its deployment.

Current research in plant disease classification is moving towards this direction. MS-Net is a lightweight CNN for plant disease detection which proved that plant diseases can be accurately detected through an efficient algorithmic approach [10].

Maize leaf disease recognition is particularly important because different leaf diseases may present visually similar symptoms, making accurate identification more challenging. Convolutional neural networks and EfficientNet-based models have been applied to maize leaf disease classification, with studies highlighting practical difficulties such as complex backgrounds and variation in disease appearance in real-world images [5], [11].

C. Problem Statement

While deep learning algorithms based on CNN have shown high levels of accuracy in classifying plant and maize leaf diseases, current algorithms have focused primarily on achieving higher levels of accuracy while neglecting computational efficiency.

For agricultural applications, particularly those intended for use outside high-performance computing environments, model parameters, storage requirements, processing time, and inference latency are relevant alongside classification performance. A model that performs well on an image-classification benchmark may nevertheless have practical limitations if its computational or storage requirements are substantially higher than those of alternative architectures.

Several existing works have highlighted the potential of lightweight CNNs in detecting diseases in plants [10]. Nevertheless, differences in datasets, preprocessing procedures, disease categories, experimental protocols, and hardware conditions make it difficult to determine how selected lightweight architectures compare when these factors are controlled.

This research therefore addresses the need for a reproducible comparative evaluation of selected lightweight CNN architectures for maize leaf disease classification. The research examines MobileNetV2, MobileNetV3-Small, and EfficientNet-B0 under a common dataset, data partition, preprocessing pipeline, training protocol, and evaluation procedure. In addition to classification performance, the study measures model parameters, model storage size, training time, inference latency, ONNX portability, and physical Android execution.

D. Aim of the Study

To comparatively evaluate lightweight convolutional neural network architectures for automated visual learning-based maize leaf disease classification, with emphasis on classification performance, computational efficiency, and deployment characteristics.

E. Specific Objectives

The study will pursue the following objectives:

1. To investigate the effectiveness of selected lightweight CNN architectures for classifying multiple maize leaf disease categories.
2. To evaluate the classification performance of the selected architectures using accuracy, precision, recall, F1-score, confusion matrices, and ROC-AUC.
3. To evaluate and compare the computational and deployment characteristics of the selected architectures using trainable parameters, model size, training time, and inference latency on desktop and Android environments

F. Research Questions

The study will address the following research questions:

1. How effectively can selected lightweight CNN architectures classify different maize leaf disease categories from leaf images?
2. What classification performance is obtained by MobileNetV2, MobileNetV3-Small, and EfficientNet-B0 when evaluated using the same dataset, preprocessing, partitioning, and testing protocol?
3. How do the selected architectures differ in terms of trainable parameters, model size, training time, and inference latency across desktop and Android environments?

G. Significance of the Research

- i. Artificial Intelligence and Computer Vision:** The research contributes experimental evidence regarding the use of lightweight CNN architectures for agricultural image classification. By evaluating multiple established architectures under a common protocol, the study provides a reproducible basis for examining the relationship between predictive performance and computational requirements.
- ii. Maize Leaf Disease Research:** The research contributes to maize leaf disease image classification by focusing specifically on four maize leaf categories: Blight, Common Rust, Gray Leaf Spot, and Healthy. The research also incorporates explicit duplicate screening and leakage-prevention procedures before model development, thereby strengthening the reliability of the experimental evaluation.
- iii. Mobile and Edge Artificial Intelligence:** The research is applicable to mobile and edge AI because model size and inference latency are examined alongside conventional classification metrics. The successful conversion of the trained models to ONNX and execution on a physical Android device provides practical evidence concerning their portability to an offline mobile inference environment.
- iv. Agricultural Applications:** An image-based disease classification system can potentially support preliminary disease recognition from maize leaf images. Nevertheless, the results of this research should be interpreted within the limits of the evaluated dataset. In

particular, the research does not claim that performance on the current dataset establishes generalization to independently collected field images.

H. Scope of the Study

This research work involves image-based classification of four different types of maize leaf: Blight, Common Rust, Gray Leaf Spot, and Healthy. The three CNN-based models under consideration include the following: MobileNetV2, MobileNetV3-Small, and EfficientNet-B0.

The study covers dataset quality assessment, duplicate screening, stratified dataset partitioning, image preprocessing, transfer learning, CNN fine-tuning, classification evaluation, computational profiling, ONNX conversion, desktop inference, and physical Android inference. The Android experiment is treated as a deployment benchmark and not as an independent field-validation study.

SECTION II

Related Work and Literature Review

A. Deep Learning and Convolutional Neural Networks for Plant Disease Detection

The use of deep learning techniques in agricultural image processing has seen a significant increase over recent years, especially in plant disease detection. Convolutional Neural Networks (CNNs) have emerged as a prominent method since it is able to learn hierarchical features from the images without the need for much manual labor in extracting features from the images. CNNs are able to learn basic features like edges and textures in lower layers and higher level features like lesions and spots in later layers.

A recent study performed a review of deep-learning methods for detecting and classifying plant diseases and found that CNN-based methods have been widely used as an important component of automated diagnosis of plant diseases [3]. In the same vein, another research performed a review of deep-learning methods for detecting plant diseases and pests and concluded that image-based deep learning can be an important direction for agricultural automation [9]. It is clear from these studies that the shift from manually designed features to learned features has greatly enhanced the capability of automated plant disease recognition systems.

A comprehensive survey of deep convolutional neural networks applied to the prediction of plant leaf diseases has been reviewed in the literature. Their work highlights the rapid growth in the use of CNNs for automated visual diagnosis and discusses key challenges such as dataset limitations, variation in image quality, illumination conditions, and the difficulty of selecting an appropriate model for different experimental environments [2]. The study further identified some of the challenges associated with deployment, such as the limitation of the datasets, variation in the environment, and the difference between the controlled laboratory images and field images. These are important challenges because even if a method works perfectly on a controlled dataset, it does not necessarily work well with variation in the lighting, background, camera properties, leaf orientation, and severity of the disease.

The advancement in complex CNN models has led to better image recognition but has increased the need for computation. The more complex the network is, the more parameters it will have and the more resources it will need in terms of memory and computation power. This presents a

challenge when implementing the system in agriculture where the target hardware could be a phone or an edge device.

B. Disease Detection in Maize Using CNN

Maize leaf disease detection has become one of the important applications of computer vision that uses deep learning since maize leaf diseases lead to significant decreases in crop yield. CNNs have been used to detect a range of diseases in maize plants, such as maize streak virus, maize lethal necrosis, blight, common rust, and other types of disease.

The study evaluated deep learning models that could help detect maize streak virus and maize lethal necrosis diseases in early stages in Tanzania [5]. It is especially interesting from the perspective of the current research due to the use of data gathered in Africa and in fields rather than lab settings. The authors have compared different approaches, such as CNNs and vision transformers, and showed that there is potential for automated detection of diseases in maize plants using deep learning techniques.

In another paper, several CNN models were considered for maize lethal necrosis and maize streak virus detection. They considered a simple CNN, EfficientNetV2-B0, EfficientNetV2-B1, LeNet-5, VGG-16, and ResNet50 [5]. EfficientNetV2-B0 obtained outstanding results regarding classification accuracy within their experimental setup. Moreover, the use of Grad-CAM was introduced in this research to increase the interpretability of their model. In this way, the efficacy of EfficientNet models in recognizing corn diseases is shown, together with the growing need for interpretability of models.

The authors proposed an enhanced EfficientNet framework for corn disease recognition [11]. This research handled the problems that may be encountered with corn diseases' pictures, such as complicated backgrounds and disease areas. It proved the efficiency of using EfficientNet models as a transfer learning method in agriculture image classification. Nevertheless, focus on classification accuracy highlights the necessity for more research to find out the computational challenges of implementing these models.

On the other hand, more recent studies on Machine Learning with Applications assessed deep-learning models for disease detection of maize plants in the field by imagery from Uganda, Tanzania, Ghana, and Namibia [12]. It compares custom CNNs, MobileNetV2,

InceptionResNetV2, Vision Transformers, and conventional machine learning models. While MobileNetV2 performed best among all classification models, the study utilized an improved YOLOv10 model for disease detection. Additionally, it created a mobile application and assessed its effectiveness on field data. Therefore, it gives a lot of weight to the assumption that lightweight models like MobileNetV2 have great potential in agricultural applications.

Another research done in Plant Methods created a mobile deep CNN model for maize leaf diseases' detection and classification [13]. Its main focus is on the fact that there is a need to automate the process of disease recognition via mobile technology, but not confine it to conventional computing systems only. Thus, the modern trend in the field is more focused on giving immediate information about disease to users.

One recent paper explored the comparison between large and lightweight CNNs for identifying maize leaf diseases [14]. This paper is very related to the current paper since the authors specifically studied the differences between large and lightweight CNNs. It confirms the need to consider both the accuracy and computation of CNNs in disease detection applications.

C. Transfer Learning for Agricultural Visual Learning

Training of a deep CNN from randomly initialized weights typically involves huge amounts of data and computations. The idea of transfer learning is training a CNN on large datasets of images and then adapting the model to solve a particular problem of agricultural classification.

The transfer learning technique has been used in many cases of plant disease recognition since the agricultural datasets tend to be much smaller than generic image datasets. The pretrained CNN gives valuable visual features that can further be trained using the plant disease images.

Transfer learning has become an important approach in agricultural visual learning because pretrained convolutional neural networks can provide effective feature representations while reducing the computational and data requirements associated with training deep models from scratch. In maize disease classification, the authors investigated an Efficient Attention Network (EANet) based on the EfficientNetV2 architecture for multi-class maize crop disease classification. Their approach incorporated attention mechanisms to enhance the extraction of discriminative disease features from maize leaf images affected by variations in background, illumination, and color characteristics. The proposed model achieved a test accuracy of 98.94%

across five maize disease classes, demonstrating the effectiveness of adapting advanced pretrained CNN architectures to agricultural visual classification tasks [15].

The EfficientNet framework has been presented to the public by the authors via the concept of compound scaling, which takes into account the depth, width and input resolution of the neural network [16]. As opposed to scaling only one of the dimensions of the network architecture, EfficientNet scales the whole architecture to achieve a better compromise between accuracy and efficiency. Such feature makes EfficientNet especially valuable for agricultural visual learning tasks.

It is therefore reasonable to use the architecture of EfficientNet as a good starting point for studying the potential of maintaining high classification accuracy with the use of computationally light solutions instead of standard deep CNNs.

D. Lightweight Convolutional Neural Networks

While deep CNNs are able to give high recognition results, their computational complexities may limit the ability to run them on limited-resource devices. Lightweight CNNs solve this problem by minimizing the number of parameters, floating operations, memory consumption, inference time, or some of those without sacrificing the ability to compete in performance.

MobileNet designs are considered some of the most significant breakthroughs in the field of lightweight CNN architectures. The author proposed MobileNetV3 that combines hardware-oriented neural architecture search and architectural design for generating architectures suitable for mobile and resource-limited settings [6]. Such architectures may be applicable to agriculture as recognizing diseases could have to be performed directly on smartphones or embedded systems.

The field of plant disease recognition has increasingly started using lightweight architectures. The author developed MS-Net a lightweight CNN for the purpose of plant disease recognition [10]. Their research was focused precisely on the problem of reducing computational resources while preserving accuracy in the process of plant disease recognition.

The authors introduced Dise-Efficient, a lightweight network model for efficient detection of plant diseases and pests [17]. In their research, the authors focused on how the trend of using bigger CNN models is shifted to lightweight models which have the potential to find an optimal point

between performance and model size. They also studied the architectural issues such as depth of convolution and kernel size, proving that architecture has its role in defining efficiency-performance tradeoff.

Another relevant study introduced FoldNet, a lightweight isotropic convolutional neural network for identifying plant diseases [18]. This work once again shows the increasing attention on designing neural architectures that would be optimized for efficient image recognition in agriculture.

In a similar way the study focused on lightweight deep-learning approaches for plant disease identification, and they described the issues caused by big network complexity and parameter numbers [19]. These results provide direct evidence of motivation for our research.

E. Lightweight CNNs for Maize Disease Detection

Although some researchers have explored lightweight CNNs for general plant disease recognition tasks, maize specific lightweight architectures have gained interest relatively recently.

Lightweight CNN architectures have shown promise for efficient maize disease classification. The authors, compared EfficientNet-B0, MobileNetV3, ShuffleNetV2, and DenseNet for maize leaf spot disease recognition, reporting that EfficientNet-B0 achieved the highest F1 score, while MobileNetV3 provided the fastest recognition speed. The findings demonstrate the importance of considering both classification performance and computational efficiency when selecting CNN architectures for practical maize disease detection applications [20].

The authors developed a lightweight network for corn leaf disease detection using the improved YOLOv8s architecture [21]. The research aimed at enhancing the efficiency of deep learning models used for disease detection without reducing their ability to accurately detect diseases. Through the alteration of the YOLOv8s architecture, a lightweight network for corn leaf disease detection under real-world conditions was developed. The research shows the emerging use of lightweight deep learning models for maize disease detection and the need to balance maize disease detection ability and computational efficiency.

F. Mobile and Edge-Based Visual Learning

Deployment environment plays a crucial role in agricultural AI systems. Cloud-based methods usually have a need to upload the image to a remote server, which may cause delays in communication and may require Internet access. Edge and mobile computing methods are thus an alternative, since they enable doing inference near the image acquisition site.

Development of lightweight deep neural networks was thus tightly connected with mobile and edge computing methods. MobileNetV3 was developed specifically for mobile computing [22]. Likewise, recent developments in agricultural research proved that lightweight disease recognition models could work well on mobile platforms [12], [13].

A comprehensive study investigated lightweight convolutional neural networks for maize leaf disease identification, highlighting the potential of computationally efficient architectures for deployment on resource-constrained devices. Their study evaluated large and lightweight CNNs and further investigated response-based knowledge distillation to improve the performance of SqueezeNet [14]. In a similar manner, the mobile-based maize disease study highlighted the increasing trend toward integration between deep-learning models and mobile applications for disease detection [13].

Such trends indicate that model evaluation must include more than just the typical classification performance metrics. For a system meant to be deployed via mobile or edge computing, model size, memory usage, computational cost, and execution time matter.

G. Evaluation of Classification and Computational Performance

Classification performance measures like accuracy, precision, recall, and F1-score still have their value in terms of multi-class disease recognition. Accuracy provides the overall proportion of correctly classified images, while precision and recall provide class-level information concerning false-positive and false-negative behavior. Macro-averaged metrics are particularly useful when class sizes differ because each class contributes equally to the final average.

For lightweight CNN evaluation, classification metrics alone are insufficient. Computational characteristics such as the number of trainable parameters, model storage size, training time, and inference latency provide additional information about practical deployment requirements.

The present study therefore treats predictive performance and computational behavior as complementary dimensions. Rather than assuming that the model with the highest classification accuracy is automatically the most appropriate architecture, the research reports the observed accuracy, macro F1-score, model parameters, ONNX storage size, training time, desktop latency, and Android latency.

H. Research Gaps

The literature review shows considerable advancement in the field of CNN-based maize leaf disease identification. There are several research gaps in this field, nonetheless.

First, existing studies use different datasets, disease categories, preprocessing procedures, data partitions, training configurations, and evaluation environments. These differences make direct cross-study comparison of reported classification performance difficult.

Second, though lightweight CNNs have been investigated for plant and maize leaf disease recognition, classification performance and computational characteristics are not always evaluated under a common experimental protocol.

Thirdly, model deployment behavior can differ from desktop behavior. Therefore, measurements obtained from a physical mobile device provide complementary information that cannot be inferred solely from parameter count or desktop inference measurements.

Fourthly, dataset quality and leakage control are important for reliable evaluation. Duplicate or highly similar images can result in overly optimistic performance if related samples are distributed across training and test subsets. Explicit duplicate screening and a reproducible data-partitioning procedure can therefore strengthen the validity of comparative evaluation.

Fifthly, field generalization remains a separate challenge. Studies using controlled or publicly available image collections cannot, by themselves, establish performance on newly collected field images. Independent field validation therefore remains necessary beyond the controlled experiment presented in this study.

I. Positioning of the Present Study

The present study addresses these issues through a controlled comparative experiment involving MobileNetV2, MobileNetV3-Small, and EfficientNet-B0.

All three architectures are evaluated using:

- i. the same cleaned four-class maize leaf dataset
- ii. the same 70:15:15 train-validation-test design
- iii. the same random seed
- iv. the same input resolution
- v. the same deterministic validation and test preprocessing
- vi. the same training-only augmentation strategy
- vii. the same transfer-learning approach
- viii. the same maximum training duration
- ix. the same independent test set
- x. the same classification and computational evaluation procedures.

The research further extends conventional classification evaluation by examining model parameters, ONNX model size, training time, desktop inference latency, Android inference latency, and consistency between the original PyTorch models and their ONNX implementations.

The primary contribution of this study is a controlled comparative evaluation of three established lightweight CNN architectures under a common experimental protocol, supplemented by desktop ONNX and physical Android deployment measurements. The framework provides experimental evidence concerning the relationship between classification performance, computational characteristics, and deployment behavior for selected lightweight CNN architectures applied to four-class maize leaf disease classification. A vital boundary of the paper's contribution is that the experiment does not establish generalization to independently collected field images. Instead, it establishes the comparative behavior of the three architectures under the defined experimental conditions while identifying independent field validation as an important direction for subsequent research.

SECTION III

Materials and Methods

A. Research Design and Experimental Framework

In this research, a controlled comparative experiment design has been used to assess the three lightweight convolutional neural network (CNN) models for automatic classification of maize leaf diseases. In this study, lightweight CNNs refers to relatively compact architectures selected for comparative evaluation under resource-conscious deployment considerations. The study evaluates three established CNN architectures, namely MobileNetV2, MobileNetV3-Small, and EfficientNet-B0. These models were selected because they provide different trade-offs between model size, computational requirements, and classification performance, making them suitable for evaluating the feasibility of CNN-based maize leaf disease classification in resource-constrained environments.

The experimental framework was designed to minimize methodological variance among the candidate architectures. Three architectures were trained and tested on the same cleaned dataset, same partitioning, image preprocessing procedure, training-only augmentation, transfer-learning approach, maximum training duration, and independent test set.

The evaluation was done based on two complementary dimensions. The first dimension was classification performance, measured using accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1-score, confusion matrices, and ROC-AUC. The second dimension was computational and deployment performance, measured using trainable parameter count, exported ONNX model size, training time, desktop inference latency, and physical Android inference latency.

The research does not introduce a new CNN architecture. MobileNetV2, MobileNetV3-Small, and EfficientNet-B0 are treated as established candidate architectures and compared under a common experimental protocol.

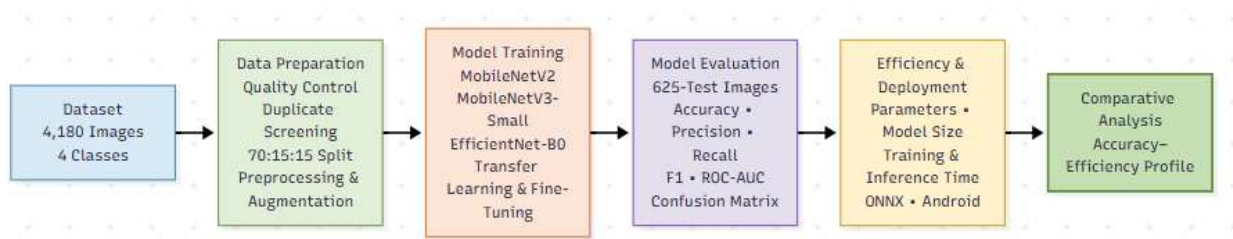


Fig. 1. Controlled comparative experimental framework for maize leaf disease classification using lightweight CNN architectures.

B. Data Set Description

The dataset used in the study was the Corn or Maize Leaf Disease Dataset, obtained from Kaggle [23]. The dataset contains 4,188 maize leaf images belonging to four classes: Healthy, Common Rust, Gray Leaf Spot, and Blight. According to the dataset description, the collection was constructed using images originating from the PlantVillage and PlantDoc datasets [24], [25], with certain images removed during dataset formation. The original authors retain rights to the respective source datasets, and their contributions are acknowledged in accordance with the dataset provider's citation requirements [23]. Before model development, the dataset was subjected to an independent verification and cleaning process. Duplicate images were identified and removed, resulting in 4,180 unique images retained for the experiments. The cleaned dataset was then partitioned into training, validation, and test subsets using a stratified procedure.

Class-wise Distribution of the original dataset is as follows: Common Rust: 1,306 Images, Blight: 1,146 Images, Healthy: 1,162 Images, Gray Leaf Spot: 574 Images.

Class Distribution Summary		
Class	Number of Images	Percentage (%)
Common Rust	1,306	31.18
Healthy	1,162	27.75
Blight	1,146	27.36

Class Distribution Summary		
Class	Number of Images	Percentage (%)
Gray Leaf Spot	574	13.71

Total images: 4,188

Table 1. Class Distribution of the Corn or Maize Leaf Disease Dataset

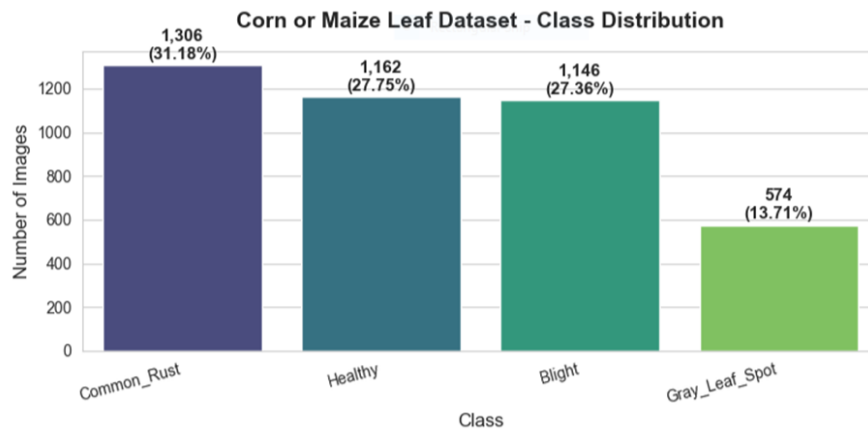


Fig. 2. Bar Chart of Class Distribution of the Corn or Maize Leaf Disease Dataset

This dataset has been created using the commonly used PlantVillage and PlantDoc Datasets [24], [25], wherein the images not useful for training were eliminated. The dataset comprises four maize leaf categories. The resulting collection contains variation in image characteristics and disease appearance. Nevertheless, we identified duplicate/conflicting samples during data-quality screening. There were 4,188 images in the initial dataset. A systematic data quality screening procedure was then conducted to detect any images that were corrupt, duplicates, and images that were nearly duplicates. There were no corrupt images detected. Eight duplicate images were quarantined, leaving the modelling dataset with 4,180 unique images.

Class	Images	Percentage (%)
Blight	1,143	27.34
Common Rust	1,305	31.22

Gray Leaf Spot	571	13.66
Healthy	1,161	27.78
Total	4,180	100.00

Table 2. Final cleaned dataset distribution table

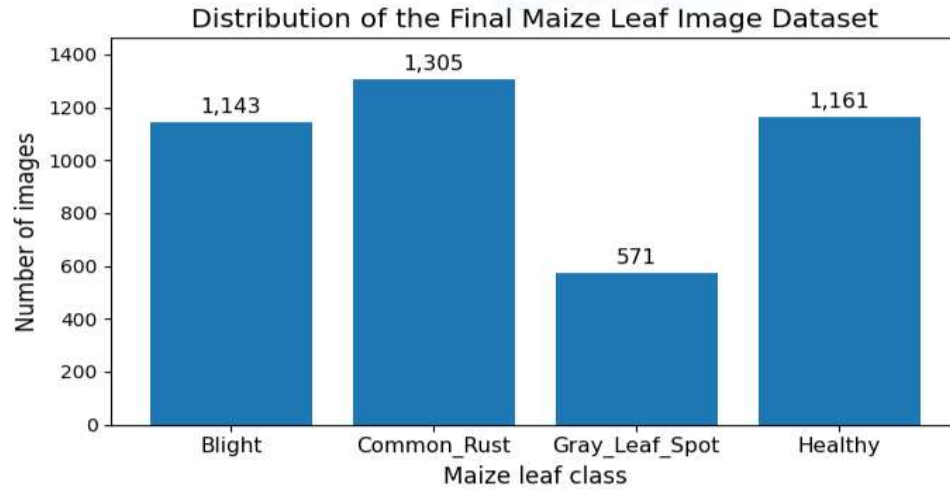


Fig. 3. Distribution of the Final Maize Leaf Image Dataset

C. Data Quality Assessment and Leakage Prevention

Data-quality control was conducted before dataset partitioning and model training. The purpose was to reduce the possibility that corrupted, duplicated, or highly similar images would influence the evaluation.

The following checks were performed:

- i. detection of corrupt or unreadable image files;
- ii. verification of image file formats;
- iii. identification of duplicate images;
- iv. identification of identical image content stored under different filenames;
- v. SHA-256 hashing for final exact-duplicate verification;
- vi. inspection of image dimensions and colour modes;
- vii. verification of class labels; and

- viii. physical verification of the final train, validation, and test directories against the generated split metadata.

Eight duplicate files were quarantined before model training. The final 4,180 image dataset contained no remaining exact SHA-256 duplicates. The duplicate control process was completed before dataset partitioning. This prevented identified exact duplicates from being distributed across different subsets and reduced the possibility of artificially inflated performance resulting from identical images appearing in both training and evaluation data.

Near-duplicate and visually similar images were taken into consideration as an issue related to the data quality. Exact duplicate screening cannot prove complete independence between all images, particularly when the source dataset does not provide complete acquisition-level metadata. Therefore, the results are interpreted as performance on the cleaned dataset rather than as evidence of complete independence of every image acquisition.

D. Dataset Partitioning

After the data-quality assessment, the resulting 4,180-image dataset was partitioned into training, validation, and test subsets using a target ratio of 70%, 15%, and 15%, respectively. In order to get the reproducible partition, a fixed random seed of 42 was used. Stratified sampling was performed jointly according to the class label and image-resolution group, where the resolution groups were defined as 256×256 pixels and non- 256×256 pixels.

The resulting partition was:

Class	Training	Validation	Test	Total
Blight	801	172	170	1,143
Common Rust	913	196	196	1,305
Gray Leaf Spot	400	86	85	571
Healthy	813	174	174	1,161
Total	2,927	628	625	4,180

Table 3. Train–validation–test partition of the dataset

The complete dataset was therefore allocated as 2,927 training images, 628 validation images, and 625 test images. The resulting proportions were approximately 70.02%, 15.02%, and 14.95%, correspondingly, with small deviations caused by integer allocation during stratified sampling.

The physical contents of the three subsets were subsequently verified against the generated split-assignment metadata, confirming that all 4,180 images were accounted for without missing files. The 625 test subset was held out exclusively for final performance evaluation and was not used during model training or model-selection decisions.

E. Image Preprocessing

A standardized preprocessing pipeline was applied to all CNN architectures to ensure that model comparisons were conducted under equivalent input conditions. First, all images were converted to RGB format to provide a consistent three-channel representation, including images originally stored in RGBA or CMYK format. The images were resized to 224×224 pixels, providing a common input resolution for all candidate architectures. This standardization takes into account the differences in the original image size detected during dataset quality assessment. Pixel values were converted to floating-point representation and scaled to the range $[0, 1]$. Subsequently, channel-wise normalization was performed using the ImageNet mean and standard deviation, with mean $\mu = [0.485, 0.456, 0.406]$ and standard deviation $\sigma = [0.229, 0.224, 0.225]$, respectively. The same deterministic resizing, RGB conversion, scaling, and normalization procedure was applied to the validation and test images to ensure that their evaluation was performed without stochastic transformations.

$$x'_c = \frac{x_c - \mu_c}{\sigma_c}, \quad c \in \{R, G, B\}$$

where

$$\mu = [0.485, 0.456, 0.406] \text{ and}$$

$$\sigma = [0.229, 0.224, 0.225].$$

F. Training-Only Data Augmentation

In order to improve the ability of the CNN models to learn discriminative features while reducing sensitivity to variations in leaf orientation, position, scale, and illumination, stochastic data augmentation was applied exclusively to the training subset. The augmentation pipeline consisted of random rotations within the range of -20° to $+20^\circ$, random horizontal flipping, random resized cropping, and random adjustments to image brightness and contrast. These transformations were generated dynamically during training rather than being permanently added to the dataset. No stochastic augmentation was used for the validation and test subsets. Therefore, validation and test images were processed only through the deterministic preprocessing pipeline, allowing model performance to be evaluated on images that had not been artificially modified. Restricting augmentation to the training subset also prevented augmented versions of evaluation images from contributing to model training and helped maintain a clear separation between model development and final performance assessment.

G. Model Architectures

Three lightweight CNN architectures were evaluated:

- i. MobileNetV2
- ii. MobileNetV3-Small
- iii. EfficientNet-B0

Each model was initialized with official ImageNet-pretrained weights. The original classification head was replaced by a new fully-connected layer with four output units corresponding to the target classes. End-to-end fine-tuning was performed for all models under identical training conditions.

- i. MobileNetV2: MobileNetV2 utilizes inverted residual blocks and linear bottlenecks to design an efficient CNN architecture. It is an appropriate reference architecture to test lightweight image classification under computational constraints.
- ii. MobileNetV3-Small: MobileNetV3-Small is an extension to the MobileNet family through architectural search techniques and mobile-friendly architecture optimizations. The

inclusion of MobileNetV3 was necessary to investigate the benefits of a more recent lightweight CNN in terms of maize leaf disease classification.

- iii. EfficientNet-B0: EfficientNet-B0 utilizes compound scaling techniques that effectively strike a balance between network depth, width, and input image size.

The candidate architectures were not treated as ablation variants of one another. Instead, they were treated as comparative lightweight CNN architectures.

H. Model Adaptation and Transfer Learning

Transfer learning was used for all three candidate architectures.

The adaptation procedure consisted of:

1. loading the ImageNet-pretrained architecture
2. replacing the original classification head
3. configuring the new head for four maize leaf classes
4. fine-tuning the adapted network using the maize leaf disease dataset
5. monitoring validation performance during training
6. retaining the model checkpoint associated with the selected validation performance

The same transfer-learning approach was applied to all three architectures to reduce differences caused by model-development procedures.

I. Training Protocol

All models were trained with the following fixed hyper-parameters to ensure reproducibility:

- i. Loss function: Cross-entropy loss
- ii. Optimizer: AdamW
- iii. Initial learning rate: 1×10^{-4}
- iv. Batch size: 32

- v. Maximum number of epochs: 30
- vi. Weight decay: 1×10^{-5}
- vii. Learning-rate schedule: ReduceLROnPlateau (factor = 0.5, patience = 5)
- viii. Early stopping: patience of 10 epochs monitored on validation loss
- ix. Random seed: 42

The models were trained using a CPU-based computing environment. The best model checkpoint was determined using validation performance rather than training accuracy alone. This procedure prevented the model-selection process from using the independent test set.

The three models were trained for a maximum of 30 epochs. Their recorded best validation accuracies were 96.18% for MobileNetV2, 97.29% for MobileNetV3-Small, and 97.61% for EfficientNet-B0. These validation results were used during model development, while the 625 image test set remained reserved for final evaluation.

J. Evaluation Metrics

The final models were evaluated using the independent test set.

The following classification measures were used:

Accuracy

Accuracy measures the proportion of test images that were correctly classified:

$$\text{Accuracy} = \frac{\text{Number of correctly classified images}}{\text{Total number of test images}}$$

Precision

Precision measures the proportion of images predicted as a particular class that actually belong to that class:

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

where TP represents true positives and FP represents false positives.

Recall

Recall measures the proportion of images belonging to a class that were correctly identified:

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

where FN represents false negatives.

F1-Score

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

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

Macro-averaged precision, recall, and F1-score were reported so that each of the four classes contributed equally to the overall metric.

Confusion Matrix

Confusion matrices were used to identify class-specific error patterns and determine which disease categories were most frequently confused.

ROC-AUC

One-versus-rest receiver operating characteristic analysis was used to evaluate the discrimination capability of the models across the four classes.

K. Computational and Model-Complexity Evaluation

Classification performance was complemented by computational measurements.

The following characteristics were recorded:

- i. trainable parameter count
- ii. exported ONNX model size

- iii. model training time
- iv. desktop inference latency
- v. Android inference latency

Parameter count was used as an indicator of model capacity and structural size. ONNX file size was used as a practical measure of the storage footprint of the exported model.

L. ONNX Export and Validation

Following model training, each selected PyTorch model was exported to the ONNX format for deployment. Structural validation confirmed that each exported model accepted an input tensor with dimensions: [1, 3, 224, 224] and produced an output tensor corresponding to the four disease classes: [1, 4].

The ONNX implementations were consequently evaluated against the original PyTorch implementations using all 625 test images. The predicted class for every test image was compared between the two implementations.

The ONNX models reproduced the PyTorch predictions exactly across the complete 625-image test set. Consequently, no difference in test-set accuracy was observed between the original PyTorch models and their ONNX counterparts.

M. Desktop Inference Evaluation

Desktop inference benchmarking was conducted using the exported ONNX models and the 625 test images. Ten warm-up runs were performed before timing. Three timed inference repetitions were then conducted for each image. Mean latency, median latency, and throughput were recorded. The desktop benchmark was used to characterize inference behavior under the tested desktop CPU environment. It was not treated as a universal measure of model speed across all computer hardware.

N. Android Deployment and Inference Evaluation

The ONNX models were integrated into an offline Android application using local inference. The application performed image preprocessing, converted images to the required 224×224 RGB

representation, applied ImageNet normalization, executed the ONNX model, and returned the predicted disease class and confidence.

The Android benchmark contained 90 model records corresponding to 30 image runs and 27 unique image names. Repeated images were retained because the purpose of this experiment was runtime and prediction-consistency analysis rather than independent accuracy estimation.

The benchmark recorded:

- a) preprocessing time
- b) model inference time
- c) total processing time.

The Android experiment was conducted on a physical device and therefore provides deployment-specific evidence. It was not treated as an independent field-validation dataset or as a replacement for the formal 625-image held-out test.

O. Prediction Consistency and Model Disagreement Analysis

Prediction consistency among the three architectures was examined using the Android benchmark and representative model outputs.

The analysis considered:

- i. cases where all three architectures produced the same predicted class
- ii. cases where two architectures agreed and one differed
- iii. cases where all three architectures produced different predictions
- iv. cases where models predicted the same class but differed substantially in confidence.

Representative disagreements were retained for qualitative analysis rather than being treated as ground-truth errors. Because model disagreement alone does not establish which prediction is correct, these cases were interpreted as evidence of differences in model decision behavior and potential areas for future expert investigation.

SECTION IV

Results and Experiments

A. Dataset Quality and Final Composition

Consequently, the data-quality assessment resulted in a final modelling dataset of 4,180 images after eight duplicate images were removed from the initial 4,188 images. The final dataset included 1,143 Blight images, 1,305 Common Rust images, 571 Gray Leaf Spot images, and 1,161 Healthy images. Common Rust represented the largest class with 31.22% of the dataset, followed by Healthy at 27.78%, Blight at 27.34%, and Gray Leaf Spot at 13.66%. Though some variation in class size remained, the class distribution provided sufficient samples for supervised training and stratified evaluation. Quality-control verification confirmed that all remaining images were readable and that no exact duplicate SHA-256 hashes remained in the modelling dataset. These procedures provided a controlled dataset for subsequent model development and evaluation.

Class	Total	Training	Validation	Test
Blight	1,143	801	172	170
Common Rust	1,305	913	196	196
Gray Leaf Spot	571	400	86	85
Healthy	1,161	813	174	174
Total	4,180	2,927	628	625

Table 4. Final dataset distribution

B. Dataset Partitioning Results

The cleaned dataset was successfully partitioned into training, validation, and test subsets using the predefined stratified sampling procedure and random seed of 42. The training subset contained 2,927 images (70.02%), while the validation and test subsets contained 628 images (15.02%) and 625 images (14.95%), respectively.

The class distributions stayed closely aligned across the three subsets, with within-class training proportions ranging from 69.96% to 70.08%. Validation proportions ranged from 14.99% to 15.06%, while test proportions ranged from 14.87% to 15.02%. Independent check of the physical split directories against the split-assignment metadata confirmed that all 4,180 images were

correctly assigned and that no files were missing. The test set was subsequently kept isolated from the model-development process and reserved for final evaluation.

C. Training and Validation Performance

The three lightweight models: MobileNetV2, MobileNetV3-Small, and EfficientNet-B0, were trained up to a maximum of 30 epochs on the same dataset splits and following the same preprocessing methodology. ImageNet-pretrained weights were initialized, and the final classification layers were adjusted for the four types of maize leaf classes. A common training time of up to 30 epochs was chosen to conduct a comparison of the three models under identical conditions.

The input images were transformed into RGB images of size 224×224 pixels, followed by conversion into tensors and normalization with ImageNet means and standard deviations. The augmentation of images during the training phase included random rotation, random horizontal flipping, random resized cropping, and brightness/contrast changes. Validation and test images were not subjected to stochastic augmentation.

Training behavior showed progressiveness in learning for all three networks during the entire 30 epochs of training. MobileNetV2 obtained its best validation accuracy of 96.18% at epoch 5, and training accuracy of this network attained 98.77% at epoch 12. The best validation accuracy of 97.29% was attained by MobileNetV3-Small at epoch 20, whereas its training accuracy was attained 99.73% at epoch 27. The best validation accuracy of 97.61% was obtained by EfficientNet-B0 at epoch 11, and training accuracy of this network attained more than 99% towards the end of the training process.

Model	Total Training Epochs	Best Validation Accuracy	Best Validation Epoch	Maximum Training Accuracy	Training Time
MobileNetV2	30	96.18%	5	98.77%	223.00 min
MobileNetV3-Small	30	97.29%	20	99.73%	188.39 min
EfficientNet-B0	30	97.61%	11	>99%	490.26 min

Table 5. Training and validation performance

Difference in training and validation accuracy shows that models learned how to classify the data successfully, but training accuracy becomes better than validation accuracy in later epochs. It proves the necessity of tracking the validation accuracy through the whole training process of 30 epochs, since training accuracy cannot be used as the only measure to choose the best model checkpoint for validation.

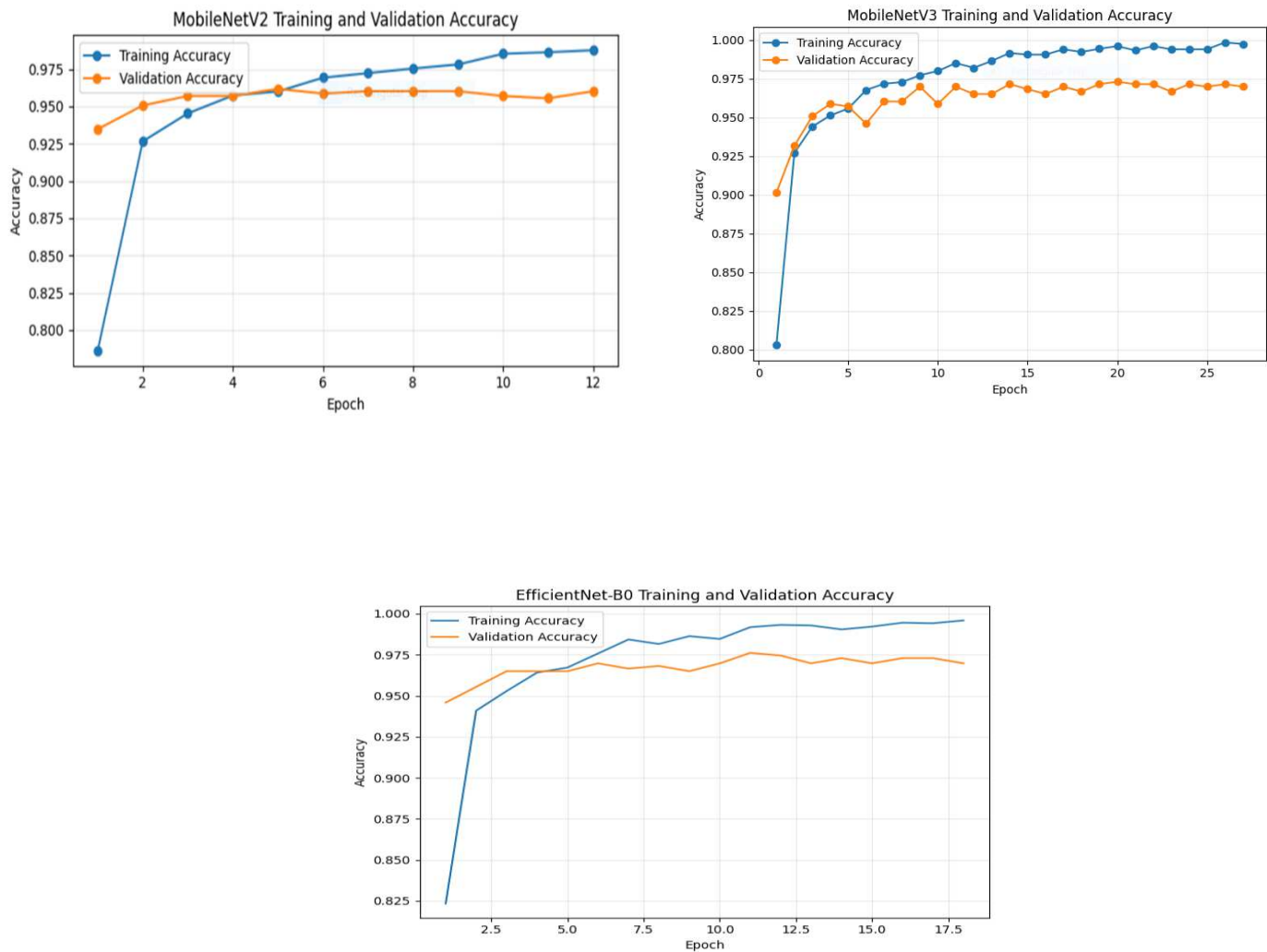


Fig. 4. Training and validation accuracy of MobileNetV2, MobileNetV3-Small, and EfficientNet-B0.

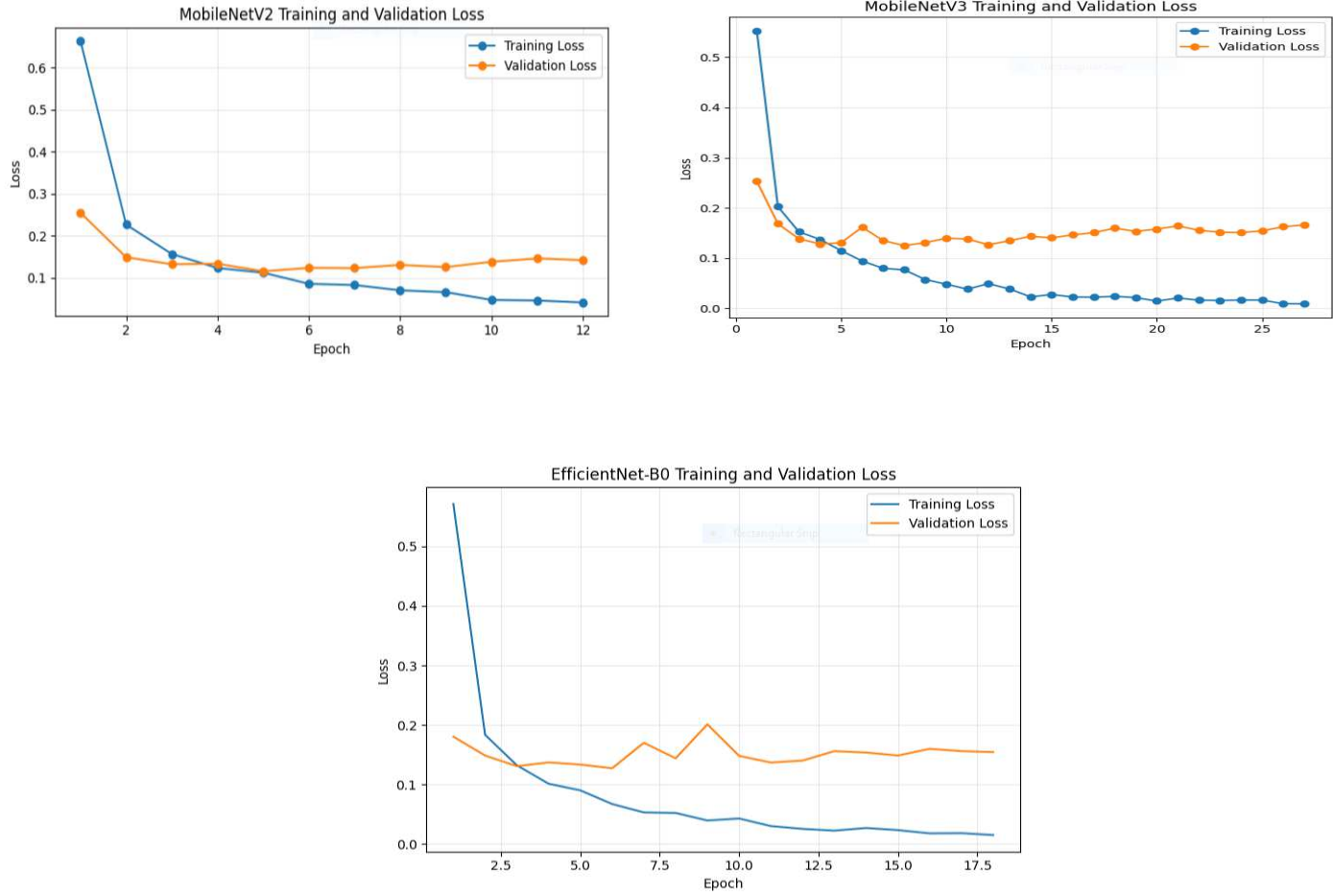


Fig. 5. Training and validation loss curves for the evaluated lightweight CNN architectures.

D. Classification Performance on the Independent Test Set

After model selection using validation performance, the three trained models were evaluated on the independent test set containing 625 images. The results show high classification performance across all three architectures.

EfficientNet-B0 had 96.96% accuracy, which meant 606 correctly classified images out of 625. MobileNetV2 had 96.80% accuracy, which means 605 correctly classified images, whereas MobileNetV3-Small had 96.00% accuracy, which meant 600 correctly classified images.

Model	Correct / 625	Accuracy	Macro Precision	Macro Recall	Macro F1	ROC-AUC
MobileNetV2	605/625	96.80%	96.36%	95.81%	96.07%	99.51%
MobileNetV3-Small	600/625	96.00%	94.95%	95.14%	95.04%	99.72%
EfficientNet-B0	606/625	96.96%	96.99%	95.83%	96.34%	99.82%

Table 6. Overall test-set performance

The variation among the three models were relatively small in terms of overall classification accuracy. EfficientNet-B0 produced 606 correct predictions, compared with 605 for MobileNetV2 and 600 for MobileNetV3-Small. It is clear from the results that the three models were able to classify the four maize leaf categories accurately within the experimental setup.

E. Class-Level Classification Results

The class-level results provide a more in-depth understanding of the model's behavior. For all three models, Healthy images were correctly classified with a recall of 100%. Also, images classified as Common Rust were classified very well. MobileNetV2 had precision, recall, and F1 score of 97.96%, MobileNetV3-Small had F1 score of 97.67%, and EfficientNet-B0 had 98.20%. The main challenge that all the models faced was the classification between Blight and Gray Leaf Spot.

Model	Class	Precision	Recall	F1-score	Support
MobileNetV2	Blight	94.22%	95.88%	95.04%	170
	Common Rust	97.96%	97.96%	97.96%	196
	Gray Leaf Spot	93.83%	89.41%	91.57%	85
	Healthy	99.43%	100.00%	99.71%	174
MobileNetV3-Small	Blight	93.06%	94.71%	93.88%	170
	Common Rust	98.95%	96.43%	97.67%	196
	Gray Leaf Spot	88.37%	89.41%	88.89%	85
	Healthy	99.43%	100.00%	99.71%	174
EfficientNet-B0	Blight	92.74%	97.65%	95.13%	170
	Common Rust	98.96%	97.45%	98.20%	196
	Gray Leaf Spot	97.40%	88.24%	92.59%	85

	Healthy	98.86%	100.00%	99.43%	174
--	---------	--------	---------	--------	-----

Table 7. Class-level precision, recall and F1-score

F. Confusion-Matrix Analysis

The confusion matrix gives further proof of the class specific behavior observed in Table 6.

For MobileNetV2, 163 of the 170 Blight images were correctly classified, while eight Gray Leaf Spot images were incorrectly classified as Blight. Conversely, four Blight images were classified as Gray Leaf Spot. For Gray Leaf Spot, 76 of the 85 images were correctly classified. For Healthy, 174 of the 174 images were correctly classified.

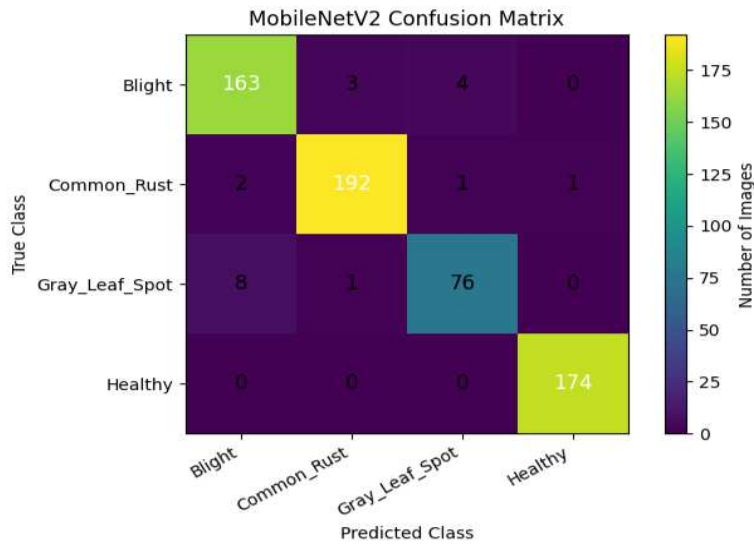


Fig. 6. Confusion matrix of MobileNetV2 on the independent 625-image test set.

MobileNetV3-Small correctly classified 161 Blight images and 76 Gray Leaf Spot images. Eight Gray Leaf Spot images were classified as Blight, while eight Blight images were also classified as Gray Leaf Spot.

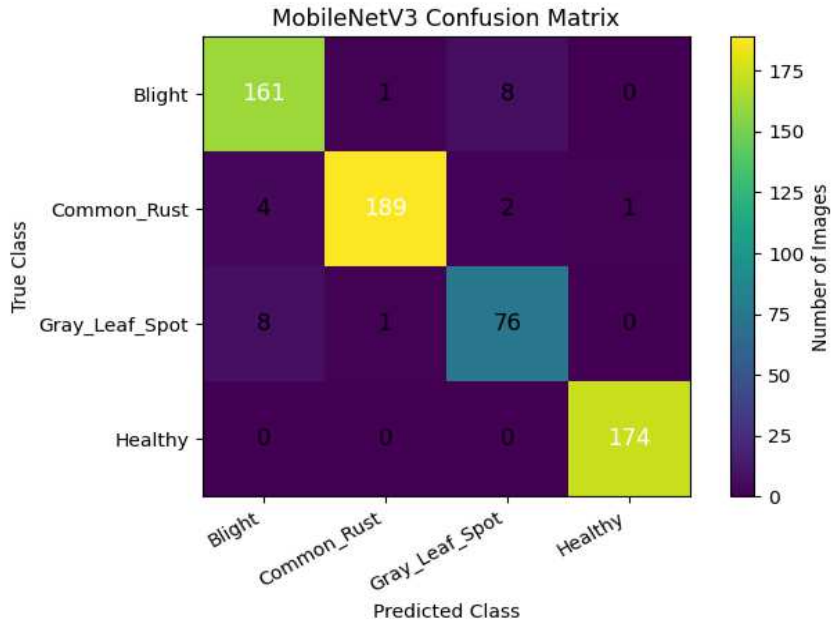


Fig. 7. Confusion matrix of MobileNetV3-Small on the independent 625-image test set.

EfficientNet-B0 correctly classified 166 Blight images and 75 Gray Leaf Spot images. Nine Gray Leaf Spot images were classified as Blight, while two Blight images were classified as Gray Leaf Spot.

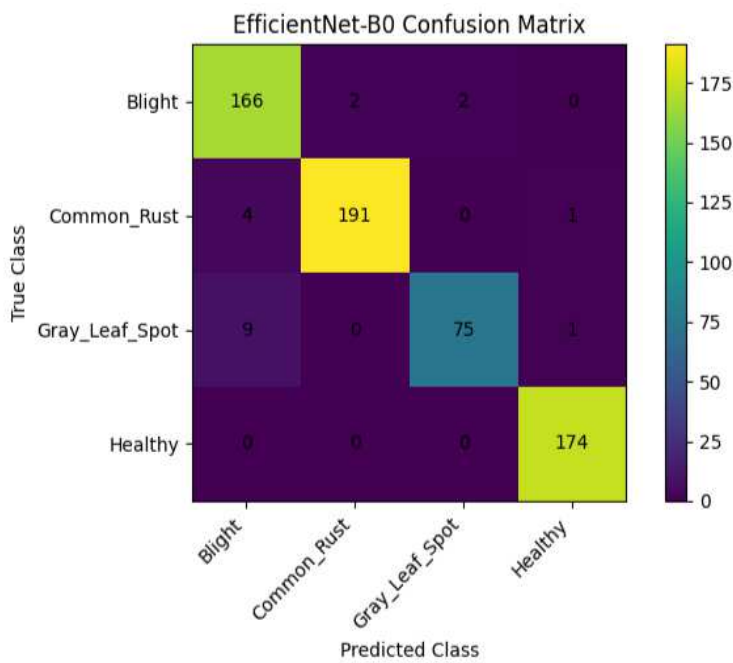


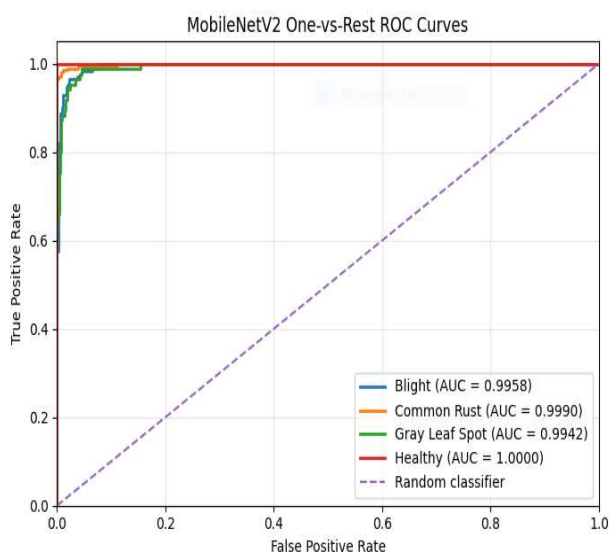
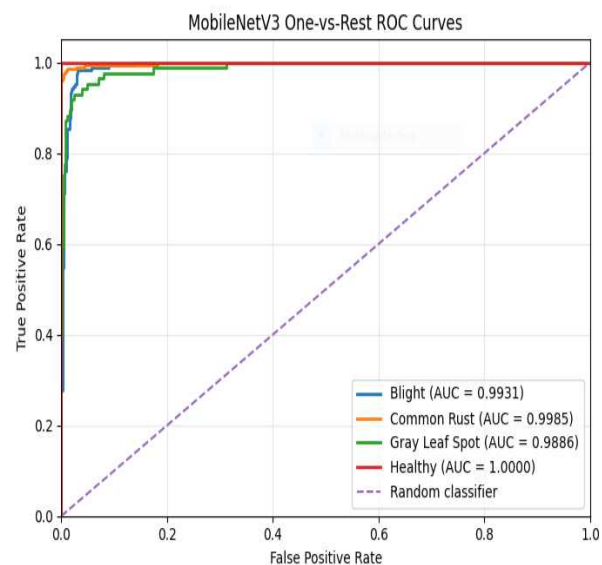
Fig. 8. Confusion matrix of EfficientNet-B0 on the independent 625-image test set.

The repeated occurrence of Blight and Gray Leaf Spot confusion across the three models is an important classification finding. Visual inspection of the misclassified samples indicated that some Blight and Gray Leaf Spot images exhibited similar overall visual characteristics, which may have contributed to the observed confusion. The recurring confusion between Blight and Gray Leaf Spot may reflect visual overlap in the evaluated images; however, establishing the underlying biological or pathological cause requires image-level analysis and expert plant-pathology verification.

G. ROC-AUC Analysis

The one-versus-rest ROC analysis produced high AUC values for all three models. The overall ROC-AUC values were 99.51% for MobileNetV2, 99.72% for MobileNetV3-Small, and 99.82% for EfficientNet-B0.

The class-specific AUC values were also high. Healthy achieved an AUC of 1.00 for all three architectures. For Gray Leaf Spot, the AUC values were 98.86%, 99.42%, and 99.68% for MobileNetV2, MobileNetV3-Small and EfficientNet-B0, respectively.



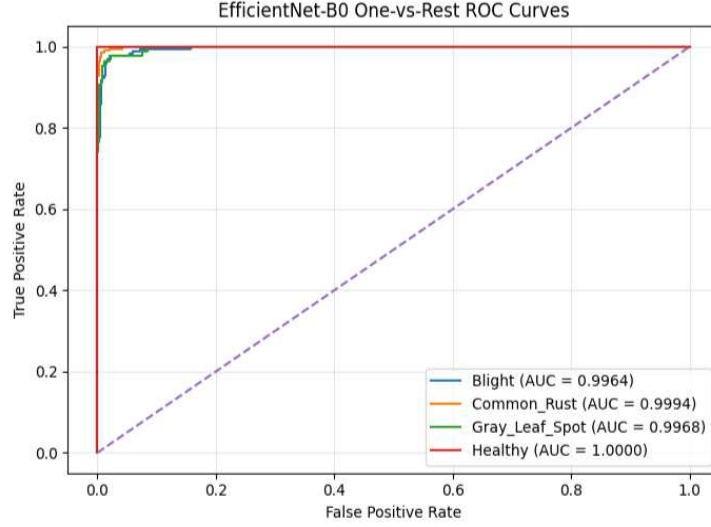


Fig. 9. One-versus-rest receiver operating characteristic curves for the three evaluated CNN models.

H. Computational Efficiency and Model Complexity

Classification performance alone does not completely describe the suitability of a CNN for low-resource and mobile deployment. This is why the following parameters, the storage size of the model, and the training time and inference latency were also evaluated.

MobileNetV3-Small contained 1,521,956 parameters, compared with 2,228,996 for MobileNetV2 and 4,012,672 for EfficientNet-B0. Consequently, MobileNetV3-Small contained approximately 31.7% fewer parameters than MobileNetV2 and 62.1% fewer than EfficientNet-B0.

The exported ONNX models had storage sizes of approximately 5.81 MB for MobileNetV3-Small, 8.48 MB for MobileNetV2 and 15.30 MB for EfficientNet-B0.

Model	Parameters	ONNX Size
MobileNetV2	2,228,996	8.48 MB
MobileNetV3-Small	1,521,956	5.81 MB
EfficientNet-B0	4,012,672	15.30 MB

Table 8. Model complexity and exported ONNX size

The training times also varied substantially. EfficientNet-B0 required approximately 490.26 minutes, compared with 223.00 minutes for MobileNetV2 and 188.39 minutes for MobileNetV3-Small. The EfficientNet-B0 training time was approximately 2.20 times that of MobileNetV2 and 2.60 times that of MobileNetV3-Small under the recorded experimental environment.

I. ONNX Model Validation and Desktop Inference

After training, the three models were exported in ONNX for deployment. Structural validation confirmed that all three exported models accepted an input tensor of $[1, 3, 224, 224]$ with floating-point values and generated an output tensor of $[1, 4]$.

The ONNX models reproduced the PyTorch test-set predictions exactly across all 625 test images. No difference in test accuracy was observed between the PyTorch and ONNX implementations.

Desktop CPU benchmarking was conducted using 625 images with 10 warm-up runs and three timed inference repetitions per image. The resulting inference latencies were:

Model	Mean Inference (ms)	Median (ms)	Throughput (images/s)
MobileNetV2	22.05	21.21	45.35
MobileNetV3-Small	7.84	6.52	127.55
EfficientNet-B0	47.24	46.36	21.17

Table 9. Desktop ONNX inference benchmark

Under this desktop CPU environment, MobileNetV3-Small recorded the lowest mean inference latency, followed by MobileNetV2 and EfficientNet-B0. It is important to note that these desktop measurements describe the specific computer environment used for benchmarking and should not be interpreted as direct measurements of smartphone performance.

J. Real-Device Android Deployment

The three ONNX models were then integrated into an Android application using React Native/Expo and ONNX Runtime. The application was configured to load the models locally rather than requiring a cloud-based inference service. The deployed application performs image preprocessing, converts the image to the required 224×224 RGB tensor, applies ImageNet normalization, executes the ONNX model and displays the resulting disease classification and confidence.

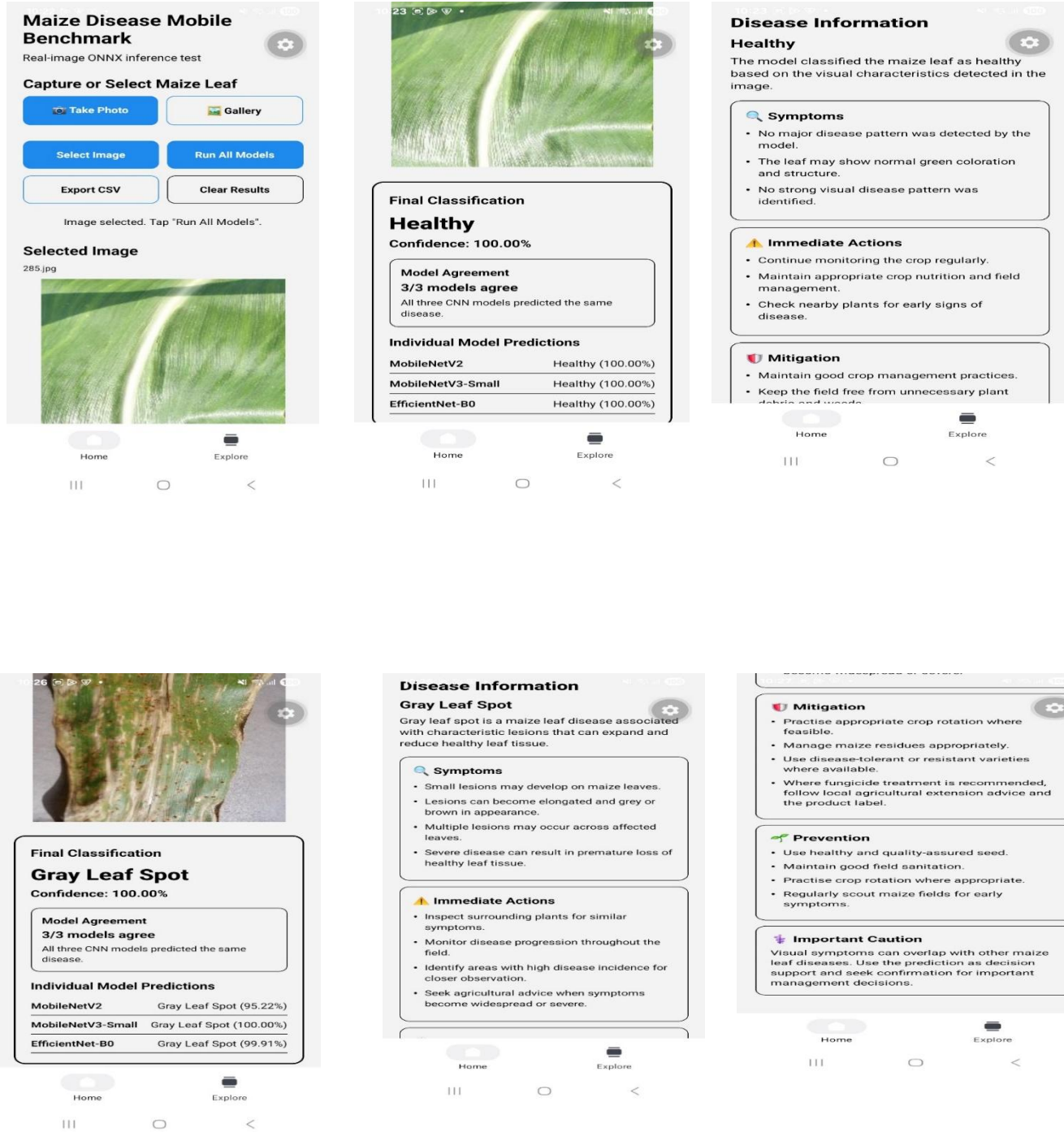


Fig. 10. Android deployment of the maize leaf disease classification system using representative test-set images: Healthy maize leaf and Gray Leaf Spot, with the corresponding classification results displayed by the application. The leaf images are from the publicly available Corn or Maize Leaf Disease Dataset [23], which was constructed using images originating from the PlantVillage and PlantDoc datasets [24], [25].

K. Android Inference Benchmark

The Android benchmark generated 90 model records corresponding to 30 image runs and 27 unique image names, with repeated images retained for runtime and prediction-consistency analysis. The 90 records represent model-level runtime observations generated from 30 image runs and should not be interpreted as 90 independent image samples.

The benchmark measured preprocessing time, model inference time and total processing time. Across the benchmark records, preprocessing accounted for approximately 458 ms per image run on average, because image resizing, JPEG rendering and tensor preparation were included in the mobile workflow.

Model	Mean Inference (ms)	Median (ms)	SD (ms)	Minimum (ms)	Maximum (ms)
MobileNetV2	47.81	45.75	5.85	43.22	71.26
MobileNetV3- Small	67.33	67.23	13.92	19.36	104.44
EfficientNet-B0	115.57	115.06	11.06	85.33	157.69

Table 10. Android real-device inference performance summary

The corresponding mean total processing times, including preprocessing, were approximately 505.93 ms for MobileNetV2, 525.46 ms for MobileNetV3-Small and 573.70 ms for EfficientNet-B0.

A notable observation is that the ordering of inference latency on the Android device varied from the desktop ONNX benchmark. MobileNetV2 recorded the lowest mean Android inference time in this benchmark, followed by MobileNetV3-Small and EfficientNet-B0. This demonstrates that desktop inference speed should not automatically be interpreted as smartphone inference speed.

The benchmark therefore provides complementary evidence: the formal 625-image experiment establishes classification performance, while the Android experiment establishes deployment behavior under the tested mobile environment.

L. Summary of Experimental Findings

The experimental results demonstrate that all three lightweight CNN architectures achieved high classification performance on the independent test set. Accuracy ranged from 96.00% to 96.96%, while macro F1-score ranged from 95.04% to 96.34%.

However, the architectures varied greatly in terms of computational characteristics. MobileNetV3-Small had the smallest parameter count and ONNX storage footprint, whereas EfficientNet-B0 had the largest parameter count and model size.

The confusion matrices displayed the principal recurring classification overlap involved Blight and Gray Leaf Spot. Common Rust and Healthy were generally classified with high precision and recall across the three models.

The successful ONNX conversion and Android deployment further demonstrated that the trained models could be transferred from the research environment into a local mobile inference workflow.

SECTION V

Discussion

A. Interpretation of Classification Performance

The experimental results show that lightweight CNN architectures are capable of delivering high classification performance when applied to the four-class maize leaf diseases problem investigated in this study. The three models produced test accuracies between 96.00% and 96.96%, demonstrating relatively similar predictive performance under the same dataset partition and preprocessing protocol.

The relatively narrow difference in accuracy is important because it indicates that model selection cannot be based solely on the highest classification accuracy. EfficientNet-B0 correctly classified 606 of the 625 test images, while MobileNetV2 correctly classified 605. This difference corresponds to only one additional correctly classified test image.

MobileNetV3-Small classified 600 of the 625 images correctly. Although its test accuracy was lower than that of the other two architectures, it retained strong overall classification performance while requiring substantially fewer parameters and a smaller exported model.

Therefore, the experimental results support an evaluation framework in which classification effectiveness and computational efficiency are considered together rather than treating accuracy as the only criterion.

B. Class-Specific Behavior

According to the class level result, it can be seen that the Healthy class was more consistent in terms of the model's performance across the three models. All the three models performed at 100% recall rate for the Healthy class.

On the other hand, the Common Rust also showed high precision and recall rates for all the three models. The Gray Leaf Spot class was observed to have low recall rate especially for the MobileNetV2 and MobileNetV3-Small. Among the three models, the EfficientNet-B0 showed high Gray Leaf Spot precision while having recall rate less than 90%.

Another observation based on the confusion matrix is the interaction between the Blight class and Gray Leaf Spot class. It can be seen that the combination of these two classes created comparatively harder classification boundary in the experiment.

C. Accuracy Efficiency Considerations

The results reveal the importance of evaluating lightweight CNNs using multiple dimensions.

EfficientNet-B0 produced the highest test accuracy at 96.96% and the highest overall ROC-AUC at 99.82%. Nevertheless, it also contained 4.01 million parameters and produced the largest ONNX model at approximately 15.30 MB.

MobileNetV2 attained 96.80% accuracy with approximately 2.23 million parameters and an 8.48 MB ONNX model.

MobileNetV3-Small attained 96.00% accuracy with approximately 1.52 million parameters and a 5.81 MB ONNX model. Its parameter count was approximately 62.1% lower than EfficientNet-B0.

These results demonstrate the central trade-off investigated by this study: increasing model capacity does not necessarily produce a proportionally larger classification benefit. In the present experiment, the difference between the highest and lowest test accuracy was less than one percentage point, while the model sizes and computational requirements differed considerably.

Therefore, deployment decisions should consider the intended operating environment, memory limitations, inference requirements and acceptable classification performance.

D. Desktop and Android Inference Behavior

The ONNX experiments confirmed that the models could be executed outside the original PyTorch training environment without changing their predictions on the 625-image test set. This provides an important reproducibility and deployment check.

Nevertheless, the desktop and Android benchmarks produced different inference-speed orderings. On the desktop CPU, MobileNetV3-Small recorded the lowest mean inference latency, while MobileNetV2 recorded the lowest mean inference latency on the tested Android device.

This variance reveals that architectural complexity alone does not completely determine practical inference latency. Runtime implementation, hardware characteristics, memory behavior, mobile processor characteristics and the surrounding preprocessing pipeline can affect observed performance.

The Android benchmark also revealed that preprocessing represented a substantial portion of the total image-processing time. Consequently, future optimization should not focus exclusively on the CNN inference operation. Image decoding, resizing, tensor preparation and other preprocessing operations should also be considered when evaluating end-to-end mobile responsiveness.

E. Prediction Consistency Across Models

The Android benchmark revealed that the three architectures did not always produce identical predictions for the same image.

The disagreements were particularly relevant for images involving Blight, Gray Leaf Spot and Common Rust. For example, image 176.jpg generated Blight predictions from both MobileNet architectures but a Common Rust prediction from EfficientNet-B0. Image 247.jpg generated a Blight prediction from MobileNetV2 and Gray Leaf Spot predictions from the other two models.

These observations are valuable for understanding model behavior. They indicate that high aggregate accuracy does not imply identical decision boundaries among architectures.

The prediction disagreements also provide useful candidates for future qualitative investigation. Images on which models disagree could be examined by agricultural experts to determine whether the images contain ambiguous disease symptoms, image-quality issues, overlapping symptoms or other characteristics that make classification difficult.

The Android benchmark should therefore be viewed as a complementary deployment analysis rather than as a replacement for the formal held-out test evaluation.

F. Implications for Mobile Deployment

The successful execution of all three ONNX models on the tested Android device demonstrates the technical feasibility of local inference under the evaluated deployment configuration

The small model sizes are particularly important for offline or low-connectivity scenarios. The MobileNetV3-Small ONNX model occupied approximately 5.81 MB, while MobileNetV2 occupied approximately 8.48 MB. EfficientNet-B0 required approximately 15.30 MB.

A mobile application can therefore perform the classification locally without transmitting the image to a remote server. This architecture may be particularly relevant to agricultural environments where internet connectivity cannot always be assumed.

However, the current Android benchmark was performed on a limited set of images and should not be interpreted as a comprehensive smartphone benchmarking study. Additional devices, repeated controlled runs and energy measurements would be required to establish broader deployment performance.

G. Research Contribution

The principal contribution of this work is not simply the achievement of high level classification accuracy. The study offers a controlled comparison of three lightweight CNN architectures using a common four-class maize leaf disease dataset, identical data partitioning, consistent preprocessing and a common evaluation protocol.

The experimental framework also extends beyond conventional classification metrics by examining model size, parameter count, ONNX portability, desktop inference and physical Android deployment.

The combination of these measurements provides a more complete picture of the relationship between predictive performance and deployment suitability.

H. Limitations

The following limitations must be acknowledged.

First, the dataset was acquired from a single principal dataset source and hence may not represent the full variability encountered under uncontrolled agricultural field conditions.

Second, even though an exact duplicate-image analysis was performed, the present study does not constitute independent field validation using newly collected images from farms.

Third, the Android benchmark contained a relatively small number of unique images compared with the 625-image formal test set. Therefore, its primary purpose was deployment and runtime evaluation rather than statistical estimation of classification accuracy.

Fourth, the Android prediction-disagreement cases were not treated as ground-truth errors. They represent differences between model outputs and require independent expert assessment if their biological correctness is to be established.

Finally, the measured mobile inference times are specific to the tested Android device and software configuration. They should therefore not be generalized to all smartphones.

SECTION VI

Conclusion and Future Work

A. Conclusion

This study presented a controlled experimental evaluation of three lightweight convolutional neural network architectures MobileNetV2, MobileNetV3-Small and EfficientNet-B0 for automated classification of maize leaf diseases.

The final modelling dataset consisted of 4,180 unique images belonging to four classes: Blight, Common Rust, Gray Leaf Spot and Healthy. Duplicates image analysis was conducted before the development of the models, and the dataset was divided into the training, validation and independent testing partitions using stratified sampling with the fixed random seed.

All three architectures achieved high classification performance on the independent 625-image test set. MobileNetV2 achieved an accuracy of 96.80%, MobileNetV3-Small achieved 96.00%, and EfficientNet-B0 achieved 96.96%. The corresponding macro F1-scores were 96.07%, 95.04% and 96.34%, respectively. The overall ROC-AUC values were also high, ranging from 99.51% to 99.82%.

The class-level analysis demonstrated particularly strong performance for Healthy and Common Rust. The most recurrent classification interaction was between Blight and Gray Leaf Spot, indicating that these classes represented the principal source of confusion in the evaluated dataset.

The successful ONNX conversion was verified by comparing predictions against the original PyTorch models. The three ONNX implementations reproduced the test-set predictions without a change in classification accuracy.

Most importantly for practical deployment, all three models were successfully executed on a physical Android device.

B. Future Work

Future research should extend the current framework in several directions.

Independent field-image validation: Future work should evaluate the models using newly collected maize leaf images from agricultural fields rather than relying solely on the existing dataset. Such

evaluation would provide stronger evidence regarding generalization to uncontrolled lighting, backgrounds, leaf orientations, camera conditions and maize varieties.

Larger mobile-device evaluation: The Android benchmark should be extended to multiple smartphone models with different processors, RAM capacities and Android versions. This would allow more robust evaluation of real-world inference latency and memory requirements.

Explainable Artificial Intelligence: Future work should incorporate explain-ability techniques such as Grad-CAM or related visual attribution methods. Such methods could help determine which leaf regions influence model predictions and could provide additional information when investigating Blight–Gray Leaf Spot disagreements.

Author Contributions

KS wrote the manuscript and JP, MN reviewed and provided guidance.

Declarations

Funding

The study did not receive any specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The authors declare that they are no conflicts of interest to publish the paper.

Consent to Publish

Not applicable.

Consent to Participate

Not applicable

Data Availability

The dataset used in this study was obtained from a publicly available Kaggle maize leaf disease dataset. According to the dataset description, the collection was compiled using images originating from the PlantVillage and PlantDoc datasets, with certain images removed during dataset formation. The original dataset contributors retain rights to the respective source datasets. The authors of the present study acknowledge these sources [23], [24], [25] as requested by the dataset provider. The Corn or Maize Leaf Disease Dataset is available at the following URL: <https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset?resource=download>. The authors of the present study independently performed dataset verification, duplicate removal, stratified data partitioning, preprocessing, model training, evaluation, ONNX validation, and Android-device benchmarking. The processed dataset split information and experimental results generated in this study are available from the corresponding author upon reasonable request.

Clinical Trial Number

Not applicable.

Ethics Declaration:

Not applicable.

Competing interests

All authors declare no competing interests.

References

- [1] R. I. Hasan, S. M. Yusuf, and L. Alzubaidi, “Review of the State of the Art of Deep Learning for Plant Diseases : A Broad Analysis and Discussion,” pp. 1–25, 2020, doi: 10.3390/plants9101302.
- [2] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, M. F. Ijaz, and M. Wo, “A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases,” 2021, doi: 10.3390/s21144749.
- [3] L. Li, S. Zhang, and B. I. N. Wang, “Plant Disease Detection and Classification by Deep Learning — A Review,” no. Ccv, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [4] A. Upadhyay, N. Singh, C. Krishna, and P. Singh, “Deep learning and computer vision in plant disease detection : a comprehensive review of techniques , models , and trends in precision agriculture,” 2025, doi: 10.1007/s10462-024-11100-x.
- [5] F. Mayo, C. Maina, M. Mgala, and N. Mduma, “Deep learning models for the early detection of maize streak virus and maize lethal necrosis diseases in Tanzania,” no. August, pp. 1–10, 2024, doi: 10.3389/frai.2024.1384709.
- [6] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, “Plant disease detection model for edge computing devices,” no. December, pp. 1–10, 2023, doi: 10.3389/fpls.2023.1308528.
- [7] R. S. Kiran, “Unified Lightweight CNN - Based Model for Multi - Crop Disease Detection and Identification,” vol. 11, no. 4, pp. 58–64, 2024.
- [8] S. Mehdipour, S. A. Mirroshandel, and S. A. Tabatabaei, “OPEN A novel lightweight hybrid CNN – ViT for maize leaf disease classification,” pp. 1–20, 2026.
- [9] J. Liu and X. Wang, “Plant diseases and pests detection based on deep learning : a review,” *Plant Methods*, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.
- [10] S. Quan, J. Wang, Z. Jia, M. Yang, and Q. Xu, “MS-Net : a novel lightweight and precise model for plant disease identification,” no. October, pp. 1–14, 2023, doi: 10.3389/fpls.2023.1276728.
- [11] J. Cai *et al.*, “Improved EfficientNet for corn disease identification,” *Front. Plant Sci.*,

- vol. 14, no. September, pp. 1–17, 2023, doi: 10.3389/fpls.2023.1224385.
- [12] J. Nakatumba-nabende, “Machine Learning with Applications Deep learning models for enhanced in-field maize leaf disease diagnosis,” *Mach. Learn. with Appl.*, vol. 20, no. May, p. 100673, 2025, doi: 10.1016/j.mlwa.2025.100673.
 - [13] G. T. Askale, A. B. Yibel, B. M. Taye, and G. D. Wubneh, “Mobile based deep CNN model for maize leaf disease detection and classification,” *Plant Methods*, vol. 21, no. 1, 2025, doi: 10.1186/s13007-025-01386-5.
 - [14] M. Haris, M. Faizal, A. Fauzi, and L. Kian, “Maize Leaf Disease Identification with Large and Lightweight Convolutional Neural Models,” vol. 9, no. March, pp. 592–598, 2025, doi: <http://dx.doi.org/10.62527/joiv.9.2.3559>.
 - [15] S. Albahli and M. Masood, “Ef fi cient attention-based CNN network (EANet) for multi-class maize crop disease classi fi cation,” no. October, pp. 1–18, 2022, doi: 10.3389/fpls.2022.1003152.
 - [16] M. Tan and Q. V Le, “EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks,” 2019.
 - [17] H. Guan, C. Fu, G. Zhang, K. Li, P. Wang, and Z. Zhu, “A lightweight model for ef fi cient identi fi cation of plant diseases and pests based on deep learning,” no. July, pp. 1–13, 2023, doi: 10.3389/fpls.2023.1227011.
 - [18] G. S. and X. Z. Wenfeng Feng, Qiushuang Song, “Lightweight Isotropic Convolutional Neural Network for Plant,” 2023, [Online]. Available: <https://doi.org/10.3390/agronomy13071849>
 - [19] J. Lu, R. Shi, J. Tong, W. Cheng, X. Ma, and X. Liu, “Lightweight Method for Plant Disease Identification Using Deep Learning,” 2023, doi: 10.32604/iasc.2023.038287.
 - [20] S. Yang *et al.*, “Classi fi cation and localization of maize leaf spot disease based on weakly supervised learning,” no. May, pp. 1–17, 2023, doi: 10.3389/fpls.2023.1128399.
 - [21] R. Li *et al.*, “Lightweight Network for Corn Leaf Disease Identification Based on Improved YOLO v8s,” 2024, doi: <https://doi.org/10.3390/agriculture14020220>.

- [22] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, “Searching for MobileNetV3,” in *Searching for MobileNetV3*, 2019, p. 11. [Online]. Available: <https://doi.org/10.1109/ICCV.2019.00140>
- [23] S. Ghose, “Corn or Maize Leaf Disease Dataset,” Kaggle. Accessed: Sep. 14, 2026. [Online]. Available: <https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset?resource=download>
- [24] G. Arun Pandian, J. Geetharamani, “Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network,” Mendeley Data. [Online]. Available: <https://data.mendeley.com/datasets/tywbtsjrjv/1>
- [25] N. Singh, Dinesh; Jain, Naman; Jain, Pratik; Kayal, Pratik; Kumawat, Saurabh; Batra, “PlantDoc: A dataset for visual plant disease detection,” in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 2020, pp. 249–253. doi: 10.1145/3371158.3371196.