

RTR_Lite_MobileNetV2: A lightweight and efficient model for plant disease detection and classification



Sangeeta Duhan ^{a,1} , Preeti Gulia ^{a,2}, Nasib Singh Gill ^{a,*³}, Ekta Narwal ^{b,4}

^a Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana 124001, India

^b Department of Mathematics, Maharshi Dayanand University, Rohtak, Haryana 124001, India

ARTICLE INFO

Keywords:

Attention mechanisms
Deep learning
Image classification
Plant disease identification
Precision agriculture

ABSTRACT

Early identification and management of plant diseases are paramount for sustaining crop health, ensuring optimal yields, and safeguarding food security in agricultural systems. Left untreated, diseases caused by fungi, bacteria, viruses, and pests can significantly diminish agricultural output, posing a threat to global food production. While recent research has explored machine learning-based techniques for early disease detection, many proposed models are resource-intensive, characterized by large model sizes, and millions of trainable parameters. Recognizing resource-constrained devices' needs, recent studies have developed lightweight models, but their shallow structure may hinder accurate disease identification. This study proposes the RTR Lite MobileNet model, an enhanced version of the original MobileNetV2 model designed for efficient deployment on resource-constrained devices. Different attention techniques, such as Squeeze-and-Excitation Networks (SENet), Efficient Channel Attention (ECA), and Triplet Attention, are added to reduce the model's computational footprint while boosting its ability to capture complicated disease patterns. Extensive experimentation validates the efficacy of RTR_Lite_MobileNet, consistently outperforming MobileNetV2 with top accuracies across multiple datasets: 99.92 % on Plant Disease, 82.00 % on PlantDoc, 97.11 % on PaddyDoctor, 90.84 % on Coffee, 100 % on Wheat, 96.78 % on Soybean, and 96.67 % on Sugarcane. Deployment on edge devices such as Raspberry Pi 4 and 5 demonstrates its computational efficiency, as evidenced by lower latency and memory consumption. Research results indicate that RTR_Lite_MobileNet is a practical and effective option for real-time plant disease diagnosis, paving the way for additional uses in agricultural monitoring and IoT applications.

1. Introduction

Achieving the Sustainable Development Goals (SDG) requires building food systems that are equitable, resilient, and sustainable. With the global population expected to reach 10 billion by 2050, agricultural advancement is necessary to reduce poverty, secure food availability, and enhance prosperity. However, the combined impact of climate change and the agricultural sector's extensive use of pesticides, fertilizers, and herbicides has intensified environmental concerns worldwide [1,2].

Plant diseases and pests cause yield losses of up to 40 % in maize,

potato, rice, soybean, and wheat crops worldwide. The annual cost of plant diseases caused by bacteria, fungi, nematodes, and viruses to the global economy is estimated at USD 220 billion [3]. Plant diseases affect agricultural yields, impact crop quality, and shorten shelf life. Plant disease also reduces the nutritional value of fruits and vegetables. On crop productivity, plant diseases have a significant impact. Therefore it is crucial to identify them in their earliest stages. Manual identification of plant diseases is difficult because it requires a certain amount of expertise and real-time observation. Additionally, if the disease identified after observation is correct or misdiagnoses happen, there is no way of knowing it. Crop health needs to be monitored to provide useful

* Corresponding author.

E-mail addresses: sangeetaduhan.rs.dcsa@mdurohtak.ac.in (S. Duhan), preeti@mdurohtak.ac.in (P. Gulia), nasib.gill@mdurohtak.ac.in (N.S. Gill), ektanarwal.math@mdurohtak.ac.in (E. Narwal).

¹ 0000-0003-2312-1211

² 0000-0001-8535-4016

³ 0000-0002-8594-4320

⁴ 0000-0001-6281-7209

information for decision-making. Therefore, it is necessary to create automated disease recognition systems that can operate accurately, quickly, and affordably. However, developing such systems is challenging and requires a significant investment of time [4]. Despite these challenges, addressing plant diseases is crucial, as they reduce the nutritional value of products and have an effect on crop quality and quantity. Early detection is therefore crucial for effectively managing plant diseases and minimizing their damage [5,6].

With the latest advancement in machine learning (ML), the Internet of Things (IoT), and computer vision, precision agriculture is now adopting cutting-edge, innovative approaches. Deep learning methods particularly convolutional neural networks (CNNs) have been instrumental in identifying key features for a variety of applications, including the detection of plant diseases. Leveraging techniques like You Only Look Once (YOLO), the real-time detection of plant diseases has become significantly more attainable [6]. Consequently, DL has gained widespread adoption, driven by the availability of large datasets, enhanced computational resources, and improved training methodologies. Standard CNN architectures such as DenseNet [7], VGG16 [8], and ResNet [9] are frequently used [10,11], researchers have developed customized CNN architectures for plant disease identification [12]. Additionally, lightweight models such as MobileNetV2 [13], ShuffleNetV2 [14], and EfficientNetLite [15] become popular as reliable options because they can be deployed on both mobile and embedded devices. Nevertheless, current improvements in plant diseases diagnosis can be considered as challenging due to several aspects. These factors include the wide range of disease types found in many crops, the advent of new diseases, and the lack of in-field data for many crops. While DL models give good results, they require abundant training data, and lightweight CNN models still struggle to generalize effectively across all crop types. There are two main challenges faced by the researcher. First is interpreting ML system decisions to anticipate potential failures, and designing efficient, lightweight CNN models that are capable of identifying a wide range of plant diseases [16].

Various studies have utilized MobileNetV2 [13] for developing solutions for automatic plant disease identification [17–19]. Some studies [20–22] have also proposed creating lightweight versions of MobileNetV2. However, the issues of generalization and result interpretation in lightweight deep learning models remain unaddressed. To tackle this issue, this study develops a lightweight MobileNetV2 model using various attention mechanisms, such as SENet and residual connections, to create the solutions.

In this study, RTR_Lite_MobileNet is developed, which is a refined lightweight variant of MobileNetV2, designed to optimize performance on resource-constrained devices. By integrating attention mechanisms such as SENet, ECA, and Triplet Attention, the proposed approach enhance feature extraction while minimizing computational demands. This work seeks to address both the generalization and interpretability challenges faced by lightweight models in plant disease identification, providing an efficient and adaptable solution suitable for real-time deployment in agricultural environments.

The main objectives of study are describe as:

1. To propose a lightweight and computationally efficient variant of the MobileNetV2 model, named RTR_Lite_MobileNet, which is trained on seven publicly available datasets and demonstrates superior results compared to the pretrained MobileNetV2 model.
2. To utilize advanced attention mechanisms such as Triplet Attention, SENet, and ECA module in the RTR Lite MobileNet model to enhance its performance by improving feature representation and classification accuracy.
3. To demonstrate how attention mechanisms can replace specific model components, reduce the model's size, and show that adding attention mechanisms to a lightweight model can lead to significant performance gains, positioning RTR_Lite_MobileNet as a reliable and efficient solution for image classification tasks.

1.1. Related works

A brief overview of recent studies that have worked on developing lightweight models is provided in Table 1. Mostly those study that have worked on improving classification results by introducing attention mechanisms for accurate plant disease identification are considered.

Panchbhai et al. [22] developed CAS-MODMOBNET model, a modified version of MobileNetV2, is developed to classify cashew fruit and nut diseases. The modification involves refining the architectural layers to reduce the overall parameter count and integrating an average pooling layer with a pool size of 3×3 to optimize feature extraction. The proposed model efficiently classified cashew fruit and nuts disease but proposed model performance on other datasets is not tested, model generalizability needs to be further test. Raj kumar et al. [23] proposed a lightweight sequential CNN architecture for diagnosing plant leaf diseases, specifically targeting on early and late blight in tomato and potato leaves. It aims to address the computational complexity of traditional CNNs by reducing parameters to approximately 70 %. While reducing computational resources demand and training time, the lightweight model struggles to capture complex hierarchical features, thus potentially impacting classification accuracy. Arya and Mishra [20] proposed a hybrid model that integrates Inception module in the last layer of MobileNetV2 architecture. They utilized pretrained model weights for transfer learning. Alkanan and Gulzar [24] present a modified MobileNetV2 architecture by adding addition layers such as Average Pooling, Flatten, Dense, Dropout, and Softmax. This, in turn, enhances the model's capacity to extract intricate features related to corn diseases, thereby improving its discriminant capabilities. Sutaji et al. [25] present a lightweight ensemble model that combines MobileNetV2 and Xception. The MobileNetV2 model is chosen for its efficiency on mobile devices, while Xception is selected for its feature extraction capabilities to improve plant disease detection. The ensemble model joins together features of both models, thus enhancing accuracy and sturdiness. Transfer learning and fine tuning are utilized to improve the model's performance. Altalak et al. [26] developed a hybrid DL approach to identify tomato plant diseases. Using three essential components—a ResNet50 for initial feature extraction, a CBAM [27] to improve the feature selection by concentrating on significant areas in the images, and support vector machines (SVM) to replace the conventional FC layer and increase classification accuracy.

Biswas et al. [12] present an energy-efficient CNN architecture for plant disease detection based on Inception and ResNet. The model minimizes computational complexity and parameter size without affecting performance. To solve the vanishing gradient problem, it uses residual connections and depth-wise separable convolution.

Chen et al. [28] proposed a model, Mobile-DANet, which aims to identify maize crop diseases using a novel network architecture. This model is constructed using DenseNet and integrates depthwise separable convolution within dense blocks. Besides that, there is a spatial and channel attention mechanism (SCAM) to learn interchannel interactions and spatial details. Initially, the model is trained on the PlantVillage dataset [29] to increase accuracy and reduce computational expenses. The focal loss function is applied in the multi-classification problem, which enhances the accuracy of disease diagnosis.

The study proposed by Shafik et al. [30] introduces two models: PDDNet-AE (Early Fusion) and PDDNet-LVE (Lead Voting Ensemble). In the PDDNet-AE, the deep features extracted from the multiple CNNs are fused by early average fusion, and for classification, a logistic regression classifier is employed. In PDDNet-LVE, the final class label of an image is determined through a majority voting method, selecting the prediction with the highest accuracy prediction from each CNN. Dai et al. [31] present the DFN-PSAN model, which is a multi-level deep information feature fusion extraction network that classifies plant disease in natural field environments. A new PSAN (Pyramidal Squeezed Attention Network) is introduced for classification and uses the YOLOv5 Backbone and Neck network as the base structure (DFN). The PSAN network

Table 1

A brief overview of recent works that have focused on developing lightweight or improved classification models for plant disease identification.

Authors	Technique	Data Pre-processing	Dataset used	Model Parameters	Accuracy	Precision	Recall	F1-Score	Limitations	Future Scope
Panchbhai et al. [22]	CAS-MODMOBNET-R And CAS -CNN	Data augmentation – Rotation, flipping	Cashew fruit and Nut images	2278,468 - CAS -CNN 2278,468 - CAS- MODMOBNET- R	97.4 %- CAS 98.7 % 98.9 %-CAS- MODMOBNET- R	98.0 %- CAS 98.9 %-CAS- MODMOBNET- R	97.0 %- CAS 100.0 %-CAS- MODMOBNET- R	97.4 %- CAS 98.7 %- CAS- MODMOBNET- R	The model's generalizability is not tested.	Dataset contain only a few images. In future work, the dataset can be increased by including more samples of disease classes.
Raj kumar et al. [23]	CNN		Tomato and Potato – PlantVillage Dataset	4200,291	98.20 %	-	-	-	Model performance on a complex dataset is not considered, and only accuracy is used for model performance evaluation.	Future work should focus on making the model more lightweight, and for model performance evaluation, different metrics can be used.
Arya and Mishra, [20]	Pre-trained mobilenetv2 and Inception module	Data Augmentation –Rotation, Brightness, Horizontal and vertical Flipping)	Rice Leaf Disease Image Samples	2502,468,	98.75 %	98.76 %	98.75 %	98.75 %	The model's generalizability is not tested.	Future research can work on improving the performance of the proposed model.
Alkanan and Gulzar [24]	MobileNetv2 + Layer Dropout		Corn Seeds Dataset	-	96.27 %	96.06 %	96.03 %	96.02 %	Proposed model's performance is low, and its generalizability is not tested.	Future research can use attention mechanisms to improve the performance and misclassification issues of the proposed model.
Sutaji and Yildiz [25]	LEMOXINET (MobileNetV2 + Xception) model.	Image resize to 224x224 pixels, normalization	Plantvillage dataset – 39 classes	26,567,310	99.10 %	99.10 %	99.02 %	99.03 %	The model is quite complex, and the generalizability of the model is not explored.	The model size is 193 MB with a trainable parameter count of 11,980,622. Future work can focus on reducing the parameters and model size to make it more lightweight.
Altalak et al. [26]	ResNet101 + CBAM + SVM	Color Conversion: RGB to BGR. Data Augmentation: Rotation, Shifting, Flipping, and Zooming	Tomato plant disease -PlantVillage Dataset		97.2 %	-	-	-	Model performance needs to be improved, and the generalizability of the model is not explored.	Attention modules give different results at different locations in the model. In future works, ablation experiments can be carried out to find the optimal position of CBAM in the proposed model.
Biswas et al. [12]	CNN	Images are cropped 100x100 pixels	Plant village Rice Dataset, Cassava dataset	3.5 million	95.17 % 99.92 % 63 %	95.11 % 99.92 % 58 %	95.17 % 99.92 % 63 %	95.11 % 99.92 % 60 %	Model performance needs to be improved.	Future works can focus on improving the model performance along with reducing the model size for its real-time use on mobile devices.
Chen et al. [28]	Mobilenetv2 + DenseNet + SCAM	Augmentation – DCGAN, random vertical or horizontal flipping,	Maize leaf image – PlantVillage Dataset		98.50 %	97.00 %	97.00 %	97.00 %	Model performance decreases when tested on a real-field dataset.	Future work can focus on improving model performance and addressing the

(continued on next page)

Table 1 (continued)

Authors	Technique	Data Pre-processing	Dataset used	Model Parameters	Accuracy	Precision	Recall	F1-Score	Limitations	Future Scope
		random angle rotation, scale transform, and colour jittering	Local maize images - 466		95.86 %	83.45 %	83.45 %	83.45 %		misclassification issue with the model.
Shafik et al. [30]	PDDNet-AE (Early Fusion), PDDNet-LVE (Major voting)	Data enhancement – noise injection, scaling, flipping, rotation, gamma correction, principal component analysis, and color augmentation.	PlantVillage Dataset – 15 classes	-	96.74 %-PDDNet-AE 97.79 %-PDDNet-LVE	-	-	-	The inclusion of multiple CNNs increases the model's complexity and interpretation.	Only the PlantVillage dataset is used to evaluate the model's performance. Future work can consider working on datasets that contain real-field images.
Dai et al., [31]	Yolov5 + PSA	Data augmentation - Gaussian filtering, non-local means denoising, and brightness/contrast adjustment	Katra-Twelve - Public dataset BARI-Sunflower - Public dataset FGVC8 – Public dataset	-	98.37 % 94.23 % 93.24 %	98.59 % 95.26 % 92.78 %	97.49 % 93.52 % 94.06 %	98.03 % 94.38 % 93.41 %	Model performance needs to be improved.	Future work can focus on using the latest YOLO variant, like YOLOv9, to see if the same modification also enhances the latest YOLO variants.
Rakibet al. [32]	Quantized CNN	-	PlantVillage Dataset - 9	26,151	98 %	98 %	98 %	98 %	The model recognises only nine classes of plant diseases.	Further research can focus on expanding the model's capabilities to identify a wider range of plant diseases. By delving into more advanced quantization techniques or utilising a combination of quantization with pruning or knowledge distillation, the model can be further optimised.
R. Maurya et al. [33]	MLP + LSTM + SVM	Random flipping, random rotation, resizing, width-and zooming	Cotton Maize PlantVillage-Tomato/ Potato/ Pepper	1041,316	98.43 % 94.27 % 97.45 %				Model performance needs to be improved.	Future directions may include testing the model on more complex datasets to test its generalizability. Models performance needs to be evaluated on different metrics.
Dheeraj et al. [34]	DenseNet121	-	PlantVillage	1.5 million	99.37 %	-	-	-	The model's generalizability is not tested.	The model size is 13.8 MB. Future work could focus on further reducing the model size and computational requirements while also maintaining accuracy.
Assaduzzaman et al. [35]	EfficientNetB0 + SE module	Resizing, scaling, horizontal flip, random rotation, color jittering	PlantVillage - Tomato	-	99.11 %	99 %	99 %	99 %	Generalizability of the model is not evaluated on other datasets	Future studies can further improve the proposed model validation accuracy, <i>(continued on next page)</i>

Table 1 (continued)

Authors	Technique	Data Pre-processing	Dataset used	Model Parameters	Accuracy	Precision	Recall	F1-Score	Limitations	Future Scope
Thai & Le, [36]	CNN + Transformer	Resizing	PlantVillage (subset)	9.2 million	-	96.88 %	96.72 %	96.80 %	The model's performance may be affected by varying lighting conditions, complex backgrounds, and hardware compatibility across diverse devices	which currently stands at 98.88 %. Future research could focus on improving robustness through dataset diversification
Li et al. [37]	Yolov8n + ATA	Rotation, scaling, cropping, and flipping	ATZD01 –Public Apricot dataset	2.79million		87.1 %	75.6 %	80.9 %	APNet's detection accuracy can still be influenced by extreme environmental variations, such as highly occluded scenes	Future research should focus on further optimizing computational efficiency and extending its application to other crops.
Joshi et al. [38]	Yolov8n	Resizing, rotation, flipping, saturation, Gaussian Blur, and Noise addition	PlantDoc – Tomato			97.75 %	93.84 %	95 %	Generalizability of the model is not evaluated on other datasets	Extending the approach to other crops and diseases by training on diverse datasets can enhance the model's versatility in precision agriculture.

emphasizes the significant areas in the image of plant disease using several convolutional layers with the pyramidal squeezed attention (PSA) mechanism. Rakib et al. [32] proposed a lightweight and energy-efficient system for plant disease identification that uses a quantized convolutional neural network (Q-CNN). The system is run on an ESP32-CAM IoT module. The CNN model is quantized to int8 format in order to reduce computational demand. The quantized model size is only 28 KB, and contains only 26,151 parameters.

R. Maurya et al., [33] suggested approach uses a combination of MLP-Mixer and Long Short-Term Memory (LSTM) models to attain a high classification performance while minimizing computational demands. Sequential data is handled in a very efficient manner by LSTM models, which excel in capturing long-range dependencies. The system uses features that are derived from pre-trained CNN models, including MobileNet, DenseNet121, DenseNet169, and DenseNet201. In last, SVM is utilized for classification tasks. Dheeraj et al. [34] put forward the LWDN (Lightweight DenseNet) model for the diagnosis of plant diseases. The LWDN model is simply the DenseNet121 model that has been trimmed down and concatenated so that it maintain the same accuracy but has lower computational requirements. During training, the model use feature fusion, transfer learning, and partial layer freezing approaches.

The Triplet Attention Mechanism (TAM) [39] has not been used as frequently in recent studies as SENet [40] and CBAM [27] mechanisms; only a few studies have included TAM in their proposed models. Li et al. [41] utilised the TAM to enhance the Yolov5 model's performance in detecting marine organisms. For this, they added the TAM module in the neck of the Yolov5 model to improve the model's feature extraction ability in capturing cross-dimensional interactions. The authors also looked at other cutting-edge attention modules, such as SENet and CBAM. However, triplet attention performed better than SENet and CBAM, showing a 1.3 % increase in mean average percision (mAP). To address the problem of detecting small targets in infrared ocean ship photos, the Ye et al., [42] presented a modified YOLOv5 model that includes a high-resolution P2 feature layer and a TAM. The P2 layer gathers detailed, low-level characteristics required for small target recognition, but it also generates noise. To remedy this, the TA module is embedded into residual blocks of the backbone network, giving additional weight to effective features while suppressing noisy ones. The incorporation of TA improves performance significantly, with greater average precision and recall rates across various target scales. In study [35], a proposed lightweight model named XSE-TomatoNet, which uses EfficientNetB0 as a backbone network to extract features at various scales, and SE module with pooling layer is also added in the network. The final classification is achieved by combining recalibrated features through fully connected layers with a softmax activation function. The MobileH-Transformer model, a novel hybrid framework that combines CNN and Transformer, is proposed in study [36]. In order to improve classification accuracy and lower computing requirements, the CNN component uses both conventional convolutional layers and a new dual convolutional block to extract highly relevant, multi-level local features. The transformer encoder processes refined feature maps with minimal attention layers to capture global dependencies efficiently. Study [37] propose APNet, a lightweight deep-learning framework, that integrates the Adaptive Threshold Algorithm (ATA) for dynamic threshold adjustment and DyHead for multi-scale feature aggregation. The ApNeck module combines shallow and deep feature integration to handle complex environments with small or overlapping targets. In study [38], the YOLOv8n model is optimized using AdamW as the optimizer and pretrained weights; the model effectively detected tomato diseases in the augmented dataset.

Even though earlier research has made great progress in creating lightweight models for plant disease detection, a number of restrictions prevent them from being directly applied to more extensive agricultural applications. The generalizability of models such as CAS-CNN [22] and study [23] is limited by their narrow focus on certain crop types or

disease stages. Similarly, while beneficial for diagnosis of corn diseases, topologies like the modified MobileNetV2 in [24] increase model complexity, making deployment on devices with limited resources less feasible. Furthermore, whereas attention techniques such as SENet [40] and CBAM [27] are integrated for feature enhancement in numerous research [21,28,31], they frequently do so without replacing computationally expensive layers, which is something we address in proposed solution. Additionally, current models are not validated across a variety of environmental circumstances, which is necessary for real-world applications. In contrast, our suggested model, RTR_Lite_MobileNet, uses sophisticated attention mechanisms like Triplet Attention, SENet, and ECA to replace more computationally demanding layers while simultaneously integrating them to capture complex information. By addressing significant shortcomings in the generalizability and flexibility of current models, this strategy aims to maximize efficiency and performance over a larger range of crops and climatic conditions.

2. Materials and methods

For effectively identifying the plant disease and addressing the challenges of the generalizability of the model, large model size and reducing the numbers of trainable parameters, this study propose Attention-fused residual lite MobileNetV2 model (RTR_Lite_MobileNet). This section explain the various techniques utilized in creating the proposed solution.

2.1. Modified squeeze and excitation (SENet) module

Hu et al. [40] proposed the Squeeze-and-Excitation (SE) block to improve the quality of representations produced by a network. They explicitly model the interdependencies among the channels of the network's convolutional features. The core idea of an SE Block is to assign different weights to each channel based on its significance. It recalibrates feature maps by explicitly modelling channel interactions and dependencies. It is widely utilized in studies [43], [44,45] for effective extract of relevant features. In the original SENet, input features are first passed through an adaptive average pooling layer, then flattened before passing it to sequential layers. But in the modified SENet, input features are parallelly passed through an adaptive average pooling and max pooling layer. The outputs from both the layers are then combined element-wise, and the resulting output follows the same process as original SENet, as shown in Fig. 1.

2.2. Efficient channel attention (ECA) module

This study also incorporated the Efficient Channel Attention (ECA) module [46]. ECA's motivation stems from recognizing that not all channels in feature maps contribute equally to the task. Some channels may capture more discriminative information, while others may contain noise or redundancy. By selectively attending to informative channels, the network enhances its discriminative power and generalization ability. ECA introduces a lightweight channel attention mechanism that directly operates on the channel dimension of feature maps.

Unlike other attention mechanisms involving costly operations like matrix multiplications, ECA employs a simple yet effective 1D convolutional operation along the channel dimension to capture channel-wise dependencies as shown in Fig. 2. This utilization of 1D convolutional operations ensures minimal computational overhead compared to more complex attention mechanisms such as self-attention, making ECA suitable for real-time applications and scenarios with limited computation resources.

2.3. RES_Block

In this study, residual attention module (RES_Block), is proposed. In RES_Block, modified SENet module is used, followed by the ECA

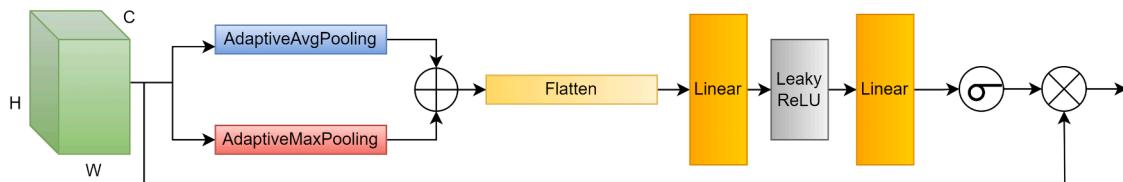


Fig. 1. Illustrate of Modified SENet Module.

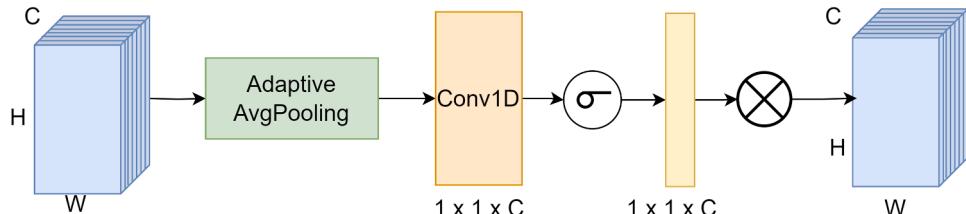


Fig. 2. Illustrate of the Efficient Channel Attention mechanism.

module, as depicted in Fig. 3. The SENet module focuses on channel dependencies. It involves a squeeze operation (global average pooling) succeeded by two fully-connected (FC) layers. The initial FC layer captures dependencies between channels, while the subsequent layer generates weights for each channel. These weights are utilized to adjust the original feature map, highlighting significant channels.

By integrating SENet modules, the network adapts channel-wise information flow, leading to better feature representation. The ECA module enhances feature representations by selectively emphasizing informative channels. Specifically, it computes channel-wise attention weights based on the global context of each channel. The residual network is used to highlight the benefits of improved feature representation, leading to enhanced accuracy and robustness. The RES_Block is as defined in Eq. (1).

$$\text{RES_Block} = x \oplus \text{ECA_Attention}(\text{SENet}(x)) \quad (1)$$

2.4. Triplet attention

Triplet Attention that is proposed in study [39], is a lightweight and effective attention mechanism that captures cross-dimensional interactions with minimal computational overhead. Previous attention mechanisms, such as SENet [40] and CBAM [27], did not consider interactions between spatial and channel dimensions, which could limit attention effectiveness. They also add significant computational complexity and learnable parameters. Triplet Attention is designed to capture interactions across different dimensions of input tensors (height, width, and channel). Triplet attention mechanism utilizes three branches that are used to calculate the attention weights by permuting

and processing the input tensor (\times) of dimension $C \times H \times W$, where C represents the channel information, H and W represent the height and width of the input tensor respectively as illustrated in Fig. 4. For each branch, the input tensor is rotated and then subjected to Z-pooling. This pooling technique reduces the size of the tensor by combining the average pooled and max pooled features across the first dimension. Subsequently, a sequence of procedures is executed to produce attention weights. This involves the use of convolutional layer, followed by batch normalization and the use of the sigmoid activation function. These branches operate in the following way:

First Branch: Explores the correlation between the H , and C dimensions. Initially, the input tensor \times is rotated 90° anti-clockwise along the H dimension. This rotation gives us a new tensor \times_1 with dimension $W \times H \times C$. After performing Z-pooling, \times_1 is transformed to a size of $(2 \times H \times C)$. Following that, a convolution operation is performed, succeeded by batch normalization and sigmoid activation to determine attention weights. These weights highlight the important features in \times_1 . The generated weights are then applied to \times_1 and then 90° clockwise rotation is performed along the H axis to preserve the original shape of \times . The Eq. (2) represents the operation of the first branch:

$$y_1 = \overline{\times_1 \odot \sigma(\omega(z(\times_1)))} \quad (2)$$

Where, z represent the Z-pooling layer, where ω represents the convolution operation with a kernel size of k , and is followed by batch normalization. Sigmoid operation is represented by symbol (σ), \odot represent element wise multiplication and the vinculum symbol represents the 90° -clockwise rotation.

Second Branch: Explores the correlation between the W and C di-

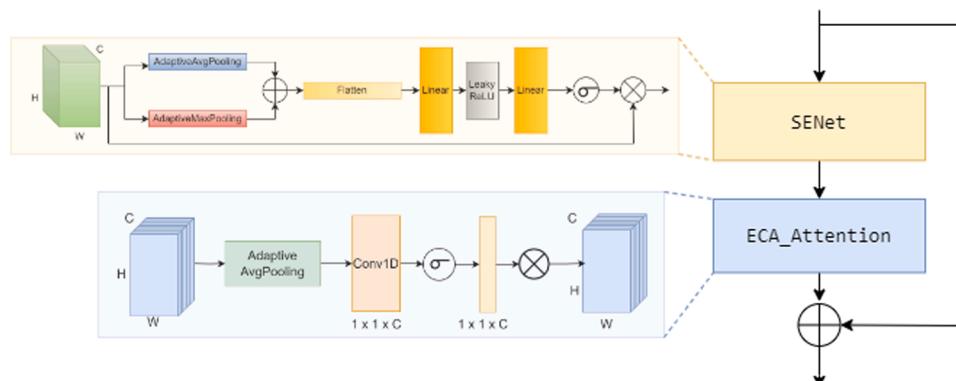


Fig. 3. Illustration of the proposed RES_Block, which has two blocks: SENet and ECA_Attention.

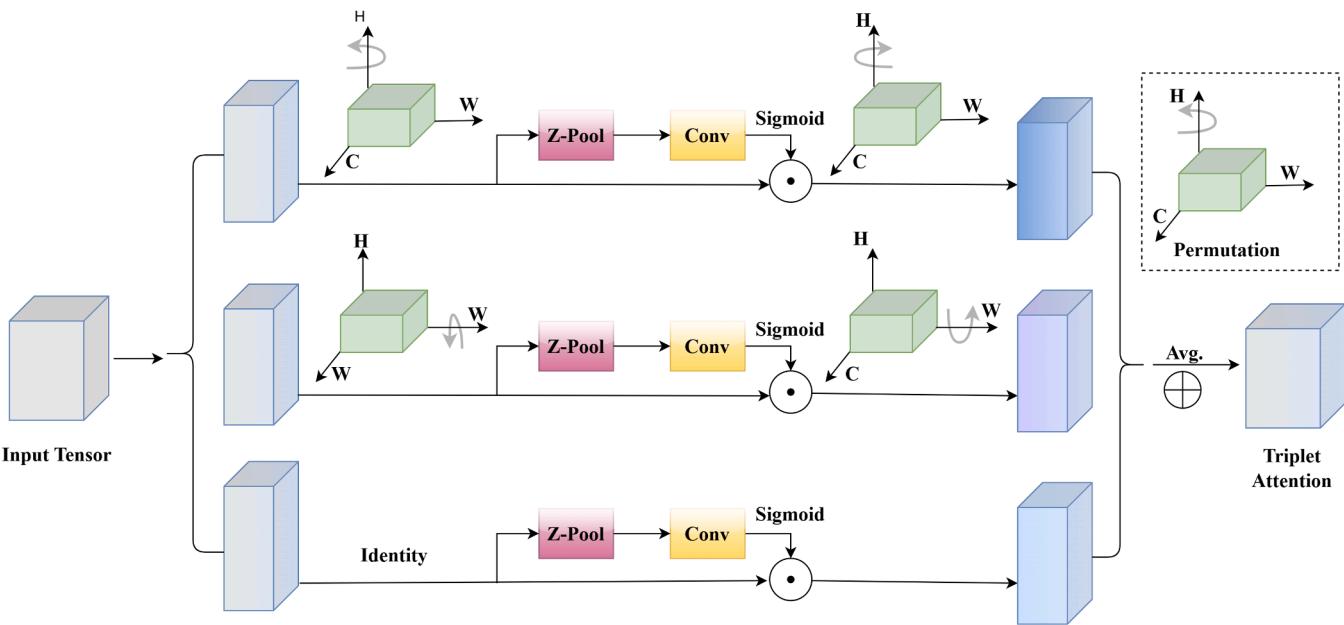


Fig. 4. Illustration of the Triplet Attention Module [39] and more details can be found in Section 2.4.

mensions. For this, the input tensor \times is rotated 90° anti-clockwise along the W dimension, resulting in the output tensor \times_2 ($H \times C \times W$). Then, Z-pooling is done, resulting in a different output tensor of shape ($2 \times C \times W$), followed by the same processing steps as the first branch. The Eq. (3) represents the operation of the second branch:

$$y_2 = \overline{\times_2} \odot \sigma(\omega(z(\times_2))) \quad (3)$$

Third Branch: Explores the correlation between the H and W dimensions. It's Apply Z-pool directly to input tensor \times , which result in output tensor ($2 \times H \times W$), followed by the same processing steps as the first branch. The Eq. (4) represents the operation of the third branch:

$$y_3 = \overline{\times} \odot \sigma(\omega(z(\times))) \quad (4)$$

At last, final attention-applied tensor is calculated by aggregating the outputs of the three branches through simple averaging, as shown in Eq. (5):

$$y = \frac{1}{3}(y_1 + y_2 + y_3) \quad (5)$$

In study [41], triplet attention is used to increase Yolov5 performance; in study [42], triplet attention is utilized to minimize noise and improve the Yolov5 model's feature extraction capabilities. Using triplet attention enables models to capture dependencies across height, width, and channels, resulting in a more robust feature representation. However, it adds convolutional operations, which may increase computational complexity and memory utilization. This may slow training and inference. The effectiveness of Triplet Attention can be influenced by the hyperparameter choices, such as the number of convolutional filters and kernel sizes used in the attention module. Using Triplet attention in any architecture may require thorough tuning and debugging to achieve optimal efficiency.

2.5. RTR_Lite MobileNet

MobileNetV2 has emerged as the preferred method for image processing and classification tasks, particularly when computational efficacy is crucial. In comparison to more complex and computationally extensive networks like AlexNet [47], VGG16, InceptionV3 [48], and ResNet, MobileNetV2 uses a linear bottleneck and an inverted residual structure to enhance convolutional processes and maintain reliable

feature extraction. It is ideal for deployment on mobile and embedded devices with constrained memory and processing capacity because of its depthwise separable convolutions, which further minimize model size and computational cost [13]. For instance, Google adopted MobileNetV2 as the backbone for the SSD [49] detector, replacing VGG16 to improve inference speed and efficiency by factorizing convolutions into depthwise and pointwise operations. More current systems, such as ShuffleNetV3 and MobileNetV3, are better but still have issues. Neural architecture search (NAS) is used by MobileNetV3 to increase accuracy; nevertheless, because it must be appropriately adjusted for unbalanced or diverse datasets, its flexibility may be constrained [50]. ShuffleNetV3, although efficient with its channel shuffle mechanism, can struggle with intricate pattern recognition in complex datasets, particularly in tasks requiring high-resolution image analysis [14]. These constraints emphasize the necessity of choosing models that achieve a balance between computational efficiency and robust feature extraction abilities, while also meeting particular application needs. MobileNetV2 has shown adaptability across numerous applications, rendering it especially useful in resource-limited settings often found in agriculture. The extensive use for identifying plant diseases arises from this adaptability. In the study [25] MobileNetV2 classification accuracy is improved by combining it with the Xception model. Study [20] combines MobileNetV2 with Inception modules to gather multi-scale information to detect complex disease patterns. To improve MobileNetV2's discriminative abilities for classifying maize diseases, the study [24] included pooling and dense layers. These studies highlight MobileNetV2's adaptability in achieving high accuracy while addressing practical challenges such as limited memory, low processing power, and the need for real-time performance.

MobileNetV2 provides an efficient backbone that supports modifications, such as the integration of attention mechanisms and layer adjustments, to create a lightweight, refined variant with minimal trade-offs in accuracy. This approach maintains an optimized balance between computational efficiency and classification accuracy, with its architecture details shown in Fig. 5(a).

To enhance feature extraction and classification capabilities, attention mechanisms like Triplet Attention, SENet, and ECA were implemented due to their complementary and distinctive characteristics. Triplet Attention captures spatial and channel-wise dependencies across multiple axes, ensuring accurate recognition of complicated patterns.

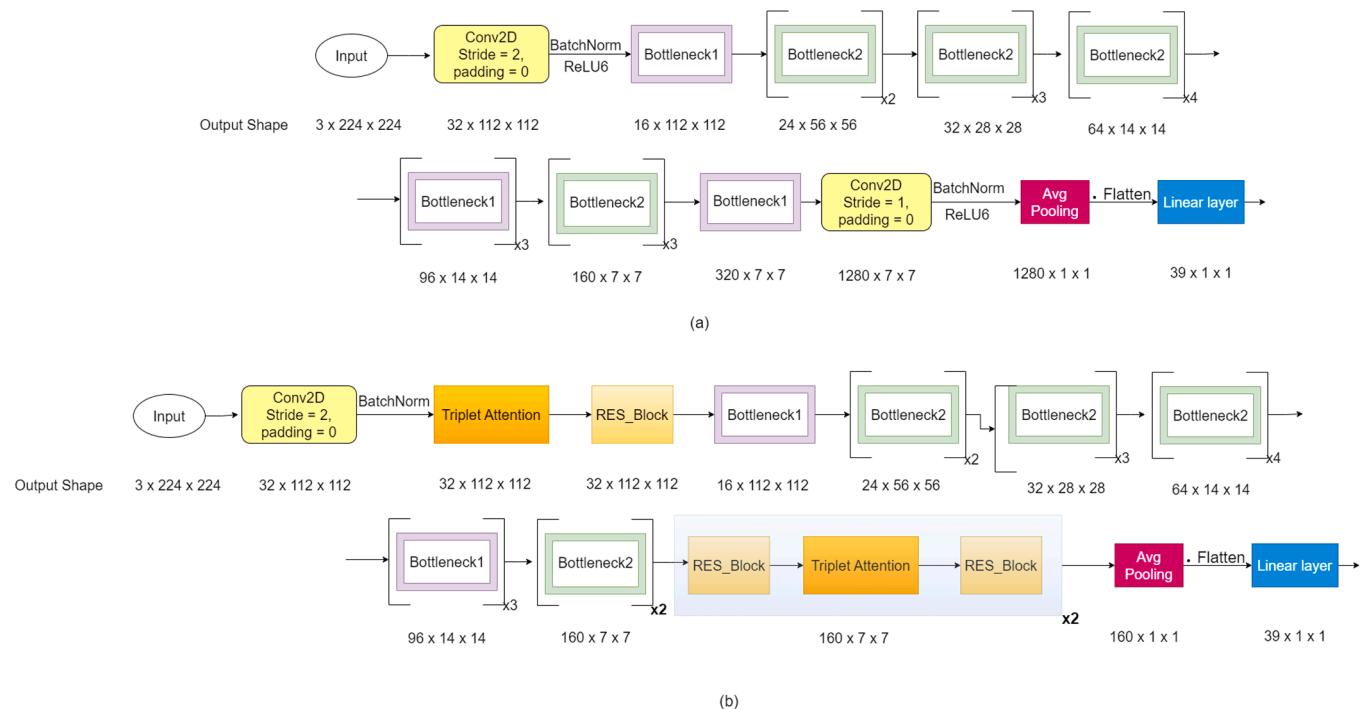


Fig. 5. Block representation of model architectures: (a) Standard MobileNetV2 architecture and (b) Proposed RTR_Lite_MobileNet architecture, highlighting the integration of advanced attention mechanisms.

SENet adjusts channel-wise feature responses to prioritize key features, while ECA enhances local cross-channel interaction with low computational cost, making it ideal for resource-constrained devices. The proposed model, as shown in Fig. 5(b), modifies the initial layer by introducing Triplet Attention and a proposed RES_Block after the first convolution layer of the standard MobileNetV2. The introduction of Triplet Attention at this stage is crucial because it enhances the model's ability to capture cross-dimensional interactions (between height, width, and channels) early in the network, thereby improving the quality of the extracted features. The RES_Block is added to provide residual connections that help maintain gradient flow, mitigate the vanishing gradient problem, and ensure better feature propagation through the network. In the proposed model, the last three layers, excluding the classifier layers, are removed. This reduces the model's parameter count, minimizing both computational cost and memory required for deployment. This makes the model more applicable to deploy on devices with limited resources. To compensate for the removal of these layers and enhance the model's capacity to learn rich features, new modules are introduced in their place. These modules comprise a RES_Block, Triplet Attention, and another RES_Block. The reason for this particular arrangement is to switch between improving spatial and channel-level attention (via Triplet Attention) and maintaining features quality and gradient flow (via RES_Block). The RES_Block enables the model's ability to learn and forward-pass the features, while Triplet Attention ensures that the model can dynamically focus on important features across different dimensions of the input. These blocks are repeated twice in the proposed model to ensure that the benefits of both Triplet Attention and RES_Block are compounded, providing a deeper and more robust feature extraction mechanism. This results in improved accuracy while keeping the model lightweight and efficient. The original MobileNetV2 model has 2273,831 total parameters and a model size of 8.69MB; the proposed model has 1050,547parameters and a model size of 4.01MB, and thus proposed model reduces the trainable parameters count and model size by 53.8 %.

3. Results

The RTR_Lite_MobileNet model performance is evaluated by utilizing seven publicly available plant disease datasets. To finalize the structure of the proposed model, various experiments are conducted, and model performance is evaluated against various metrics. The following sections contain details regarding the datasets, evaluation measures, experimental setup, and performance results obtained during the experimentation. Furthermore, the suggested model's explainability is assessed using gradient-weighted class activation maps (Grad-CAMs) [51].

3.1. Dataset and data preressing

To enhance data diversity and model robustness, this study utilizes seven publicly available datasets, namely Plant Disease Dataset [52], Coffee [53], Wheat [54], Soybean [55], Sugarcane [56], PlantDoc [57], and PaddyDoctor [58], containing images of various crops and diseases from different geographical regions. With great diversity in size and classification distribution, these databases cover a wide range of crops, diseases, and geographical areas. The datasets consist of the smaller, balanced Sugarcane dataset with 100 images across three classes and the large, imbalanced Plant Disease dataset [52] with 61,486 images over 39 classes. This variety in size and composition helps the model to learn from many agricultural settings, disease patterns, and crop varieties, hence improving its generalization and adaptability. All images were resized to 224×224 pixels to ensure consistency during training. The datasets were divided into training, validation, and testing sets in an 8:1:1 ratio, facilitating a comprehensive evaluation of the model across diverse scenarios while helping to mitigate overfitting.

Many augmentation approaches were utilized to address class imbalances and increase dataset diversity. These included random rotations of up to 30 degrees to simulate different observation angles, color jittering to adjust brightness, contrast, saturation, and hue to improve resistance against lighting variations, and random horizontal flipping to account for orientation variability. Incorporating scaling, shearing, and translations, these random affine transformations further enhance the spatial variety of the datasets. These augmentation methods were

iteratively applied on underrepresented classes until their image counts matched the largest class, therefore guaranteeing balanced datasets. Given its already balanced class distribution, the Sugarcane dataset did not require augmentation. Fig. 6 illustrates the class distributions of the datasets before and after augmentation. For each dataset, the class indexing displayed in the images is consistently used in the confusion matrices.

Fig. 7 showcases sample images from the datasets used in this study, highlighting the diversity of crop types, disease manifestations, and image characteristics. The PlantDoc dataset was specifically incorporated to evaluate the model's robustness in difficult scenarios, featuring real-world images that encompass background noise, occlusion, and varying lighting conditions. This study leverages multiple publicly available datasets, many of which include images captured in real-world agricultural settings, providing a foundation for evaluating the model's performance under practical conditions. However, despite this diversity, further validation through extensive field trials is essential to ensure robustness across varying environmental and agricultural scenarios. Future research will focus on real-life field testing to enhance the model's applicability and reliability in dynamic agricultural environments.

3.2. Evaluation metrics

The performance of the model is assessed using commonly used classification metrics, such as accuracy (Eq. 6), precision (Eq. 7), recall (Eq. 8), F1-score (Eq. 9), Area under the Receiver Operating Curve (ROC), and Cohen's kappa score. In evaluating classification techniques, it is essential to have a precise understanding of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These terms represent the different outcomes of a classification model: TP depicts the ability of the model to predict a positive class as positive, TN depicts the

model's ability to correctly predict the negative class as negative, FP depicts the case when the model wrongly predicts an instance as positive, and FN depicts the case when the model categorizes the instance as negative when the instance is actually positive [59]. These measures are computed as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (6)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (7)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

$$\text{F1Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

The ROC curve covers is known as the Area under the curve, or AUC. Plotting the true positive rate (TPR) (refer Eq. 10) and false positive rate (FPR) (refer Eq. 11) yields the ROC.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (11)$$

Cohen's kappa is a statistical measure that assesses the degree of agreement between two raters or evaluators, surpassing what might be anticipated by random chance. This approach is especially valuable for assessing the consistency among different raters in classification tasks. In machine learning, Cohen's kappa can be used to rate how well a classification model works by treating the real labels and the model's predictions as two different raters. The kappa statistic is represented in

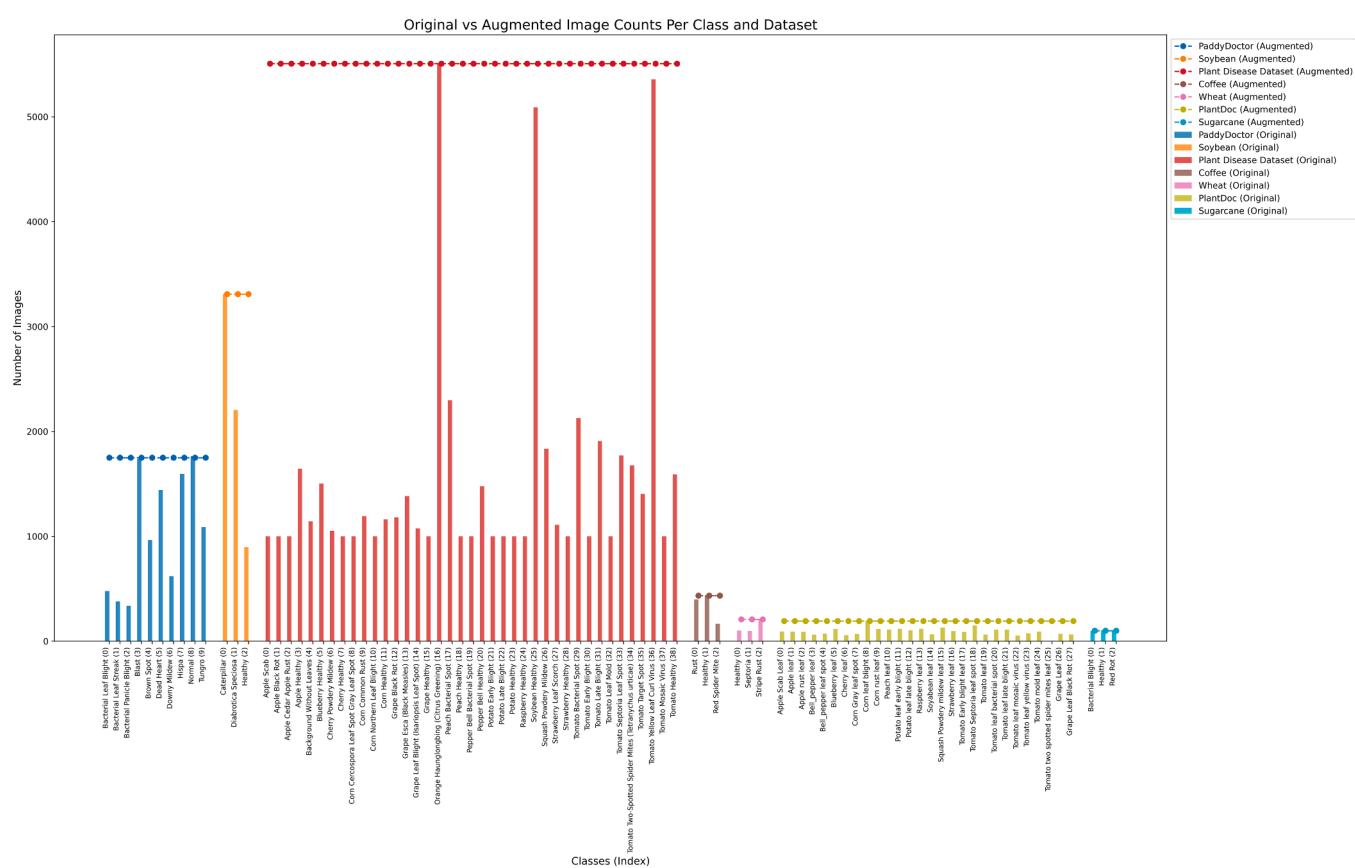




Fig. 7. Sample Images from the Datasets (a) Plant Disease Dataset [52]; (b) Coffee [53]; (c) Wheat [54]; (d) Soybean [55]; (e) Sugarcane [56]; (f) PlantDoc Dataset [57]; (g) PaddyDoctor [58].

Eq. 12.

$$\kappa = \frac{(p_o - p_e)}{(1 - p_e)} \quad (12)$$

Here, p_o refers to the observed agreement probability, and p_e is the expected agreement probability.

Each metric gives a distinct perspective of how well the model is doing. Although accuracy is a prominent metric, it can be misleading in the case of imbalanced datasets since it may overlook correct predictions in minority classes. The precision and recall metrics are critical for assessing the model's performance in terms of FP and FN. The F1-score, determined as the harmonic mean of precision and recall, plays a vital role in handling imbalanced datasets by effectively balancing these two metrics. The AUC assesses how well the model can distinguish between different classes at various thresholds, which makes it useful for both binary and multi-class situations. Finally, Cohen's kappa evaluates the correlation between the model's predictions and the true labels, taking into consideration the possibility of chance agreement and offering a more detailed measure of reliability.

3.3. Experiment setup

The study utilized the CUDA Toolkit version 12.4 and the PyTorch deep learning framework. With Python 3.10.13 as the main programming language, Jupyter Notebook was utilized as the development

platform. The experiments were carried out on an NVIDIA GeForce GTX 1080 Ti utilizing Driver Version 550.90.07 and CUDA Version 12.4. The memory capacity of the GPU was 11264MiB. Training was conducted for 30 epochs using a batch size of 32. Various optimizers, such as Adam with an initial learning rate of 0.001, were employed. The Cross-EntropyLoss function was used as the loss function. For improved training efficiency, a learning rate decay of 10 % was implemented every 5 epochs.

3.4. Comparative analysis with state-of-art models

To evaluate the model performance against other models, the Plant Disease Dataset [52] is used. **Table 2** presents a comprehensive comparison between different pretrained models like ResNet101 [8], EfficientNet [60], ShuffleNetV2 [14], ViT_B_16 [61], MobileNetV2 [13], MobileNetV3 [50] and RTR_Lite_MobileNet.

Table 2 presents a performance comparison of parameters such as computational efficiency, model size, and prediction accuracy of various architectures, including the proposed RTR_Lite_MobileNet. The visualization of the model's performance on the test dataset is presented in **Fig. 8**. As shown in **Fig. 8**, the proposed model outperforms other models in all metrics considered.

The RTR_Lite_MobileNet outperforms all the mentioned models in terms of accuracy, including MobileNetV3 (99.80 %), ResNet101 (99.69 %), and EfficientNet_V2_S (99.82 %). Moreover, the training loss

Table 2

Performance comparison of various Models on Plant Disease Dataset.

Model	Model Size (in MB)	Total Trainable parameters	Loss	Accuracy	Precision	Recall	F1-score	AUC-Score	Kappa Score	GFLOPS
ResNet101 [8]	162.43	42,580,071	0.0194	99.69 %	99.62 %	99.64 %	99.63 %	99.98 %	99.68 %	15.72
EfficientNet_V2_S [56]	77.16	20,227,447	0.0083	99.82 %	99.74 %	99.77 %	99.75 %	100.00 %	99.81 %	35.22
EfficientNetlite [60]	13.05	3,420,967	0.0091	99.80 %	99.78 %	99.73 %	99.75 %	100.00 %	99.80 %	0.745
ShuffleNet_V2 [14]	4.93	1,293,579	0.0118	99.72 %	99.65 %	99.62 %	99.63 %	100.00 %	99.71 %	0.0876
ViT_B_16 [57]	328	86,567,656	0.0100	99.77 %	99.72 %	99.73 %	99.72 %	100.00 %	99.76 %	35.22
MobileNetV2 [13]	8.69	2,273,831	0.0139	99.66 %	99.55 %	99.53 %	99.53 %	100.00 %	99.64 %	0.638
MobileNetV3 [50]	3.1	1,528,106	2.7094	99.80 %	99.80 %	99.79 %	99.80 %	100.00 %	99.83 %	0.12
RTR_Lite_MobileNet	4.01	1,050,547	0.0054	99.89 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %	0.526

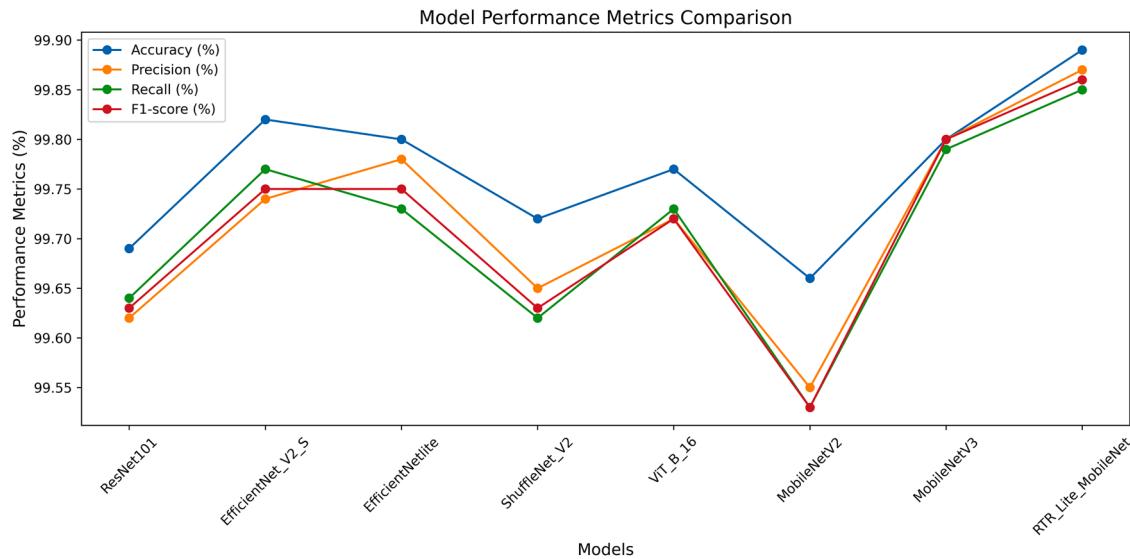


Fig. 8. Model's performance on Test dataset of Plant Disease Dataset.

achieved by the proposed model is lowest among all (0.0054), indicating better optimization and training convergence. Impressive statistics show that classification was highly accurate, with very few false positives and negatives: an F1-score of 99.86 %, precision of 99.87 %, and recall of 99.85 %. Although Cohen's Kappa score reached 99.88 %, indicating near-perfect agreement between predictions and actual labels, the 100 % AUC score validates the model's ability to identify the diseases with high accuracy. An important metric for assessing computational efficiency is the FLOP count, which measures the floating-point operations required during inference. The compact model size of 4.01 MB and a FLOP count of 0.526 GFLOPs highlight the efficient architecture of RTR_Lite_MobileNet, making it ideal for deployment on low-power devices. Even when larger models like ViT_B_16 (35.22 GFLOPs) and

ResNet101 (15.72 GFLOPs) attain competitive accuracy, their significantly higher computational costs make them less useful in situations with limited resources.

Compared to RTR_Lite_MobileNet, MobileNetV3 (0.12 GFLOPs) and ShuffleNet_V2 (0.0876 GFLOPs) exhibit lower FLOPs count, indicating low computational complexity. However, the classification accuracy and robustness of the proposed model consistently outperform current models across several criteria, particularly in real-world situations where balanced performance is necessary. RTR_Lite_MobileNet's architectural improvements significantly enhance performance metrics such as accuracy, precision, and recall, making the trade-off in computational complexity justifiable despite having more FLOPs than MobileNetV3 and ShuffleNet_V2. Future developments in model architecture may

Table 3

Performance Comparison of the RTR_Lite_MobileNet with the Standard MobileNetV2 Model [13] on Different Datasets.

Dataset	Augmentation	Model	Parameters	Loss	Accuracy	Precision	Recall	F1_score	Roc Curve	Kappa Score
Plant Disease Dataset [52]	Without Augmentation	MobileNetV2	2273,831	0.0139	99.66 %	99.55 %	99.53 %	99.53 %	100.00 %	99.64 %
	Augmentation	RTR_Lite_MobileNet	1050,547	0.0054	99.89 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %
	With Augmentation	MobileNetV2	2,273,831	0.0033	99.87 %	99.87 %	99.87 %	99.87 %	100.00 %	99.87 %
Coffee [53]	Without Augmentation	RTR_Lite_MobileNet	1,050,547	0.0032	99.92 %	99.92 %	99.92 %	99.92 %	100.00 %	99.92 %
	With Augmentation	MobileNetV2	2,227,715	0.3952	88.55 %	87.96 %	87.74 %	87.79 %	95.93 %	82.64 %
	Without Augmentation	RTR_Lite_MobileNet	1044,751	0.3858	90.84 %	90.26 %	90.32 %	90.23 %	95.50 %	86.14 %
Wheat [54]	Without Augmentation	MobileNetV2	2,227,715	0.0496	97.62 %	96.30 %	98.55 %	97.30 %	99.92 %	96.06 %
	With Augmentation	RTR_Lite_MobileNet	1,032,003	0.0594	97.62 %	98.48 %	97.62 %	97.99 %	100.00 %	96.08 %
	Without Augmentation	MobileNetV2	2,227,715	0.0252	98.41 %	97.62 %	98.25 %	97.86 %	99.80 %	97.46 %
Soybean [55]	Without Augmentation	RTR_Lite_MobileNet	1,032,003	0.0167	100 %	100 %				
	With Augmentation	MobileNetV2	2,227,715	0.3003	92.04 %	90.68 %	91.81 %	91.22 %	98.46 %	86.56 %
	Without Augmentation	RTR_Lite_MobileNet	1,044,751	0.2834	94.07 %	93.10 %	94.61 %	93.82 %	98.67 %	89.99 %
Sugarcane [56]	Without Augmentation	MobileNetV2	2,227,715	0.1711	95.88 %	95.85 %	95.83 %	95.83 %	99.28 %	93.80 %
	With Augmentation	RTR_Lite_MobileNet	1,044,751	0.1243	96.78 %	96.78 %	96.73 %	96.75 %	99.48 %	95.16 %
	Without Augmentation	MobileNetV2	2,227,715	0.1222	93.33 %	93.33 %	93.73 %	93.00 %	99.54 %	89.86 %
PlantDoc Dataset [57]	Without Augmentation	RTR_Lite_MobileNet	1,032,003	0.2260	96.67 %	94.44 %	96.67 %	95.22 %	98.63 %	94.59 %
	With Augmentation	MobileNetV2	-	-	-	-	-	-	-	-
	Without Augmentation	RTR_Lite_MobileNet	-	-	-	-	-	-	-	-
PaddyDoctor [58]	Without Augmentation	MobileNetV2	2,259,740	1.2854	61.24 %	60.94 %	59.87 %	59.17 %	95.82 %	59.58 %
	With Augmentation	RTR_Lite_MobileNet	1,048,776	1.1796	64.73 %	65.26 %	62.88 %	62.82 %	96.10 %	63.18 %
	Without Augmentation	MobileNetV2	2,259,740	0.8157	78.48 %	78.40 %	78.19 %	77.96 %	98.55 %	77.65 %
	With Augmentation	RTR_Lite_MobileNet	1,048,776	0.6416	82.00 %	82.31 %	81.55 %	81.42 %	98.98 %	81.32 %
	Without Augmentation	MobileNetV2	2,236,682	0.1636	96.45 %	96.76 %	95.57 %	96.13 %	99.76 %	95.93 %
	With Augmentation	RTR_Lite_MobileNet	1,045,878	0.1110	96.92 %	96.61 %	96.24 %	96.42 %	99.64 %	96.47 %
	Without Augmentation	MobileNetV2	2,236,682	0.1242	96.43 %	96.51 %	96.43 %	96.45 %	99.89 %	96.03 %

(-) indicate augmentation not applied

focus on reducing the FLOP count without compromising accuracy, enabling even more efficient deployment on ultra-low-power computers. Such advancements will expand the model's utility in real-time agricultural scenarios, where precise and computationally efficient plant disease identification is crucial.

3.5. Test of generalizability

Seven datasets discussed in [Section 3.1](#) were used to evaluate the performance and robustness of the proposed RTR_Lite_MobileNet model in various scenarios. The model's effectiveness, with and without data augmentation, was compared to the standard MobileNetV2 model to assess improvements in accuracy, loss, and prediction reliability. The results are presented in [Table 3](#).

The findings in [Table 3](#) demonstrate the suggested RTR Lite MobileNet model's higher performance in terms of accuracy, loss, and other evaluation criteria, while preserving a lightweight architecture with fewer parameters, making it suitable for resource-constrained applications. The corresponding training and validation accuracy, along with the loss values for augmented and non-augmented data, are visually represented in [Fig. 9](#).

In the Plant Disease Dataset [52], RTR_Lite_MobileNet outperforms MobileNetV2 by 0.23 %, attains an outstanding accuracy of 99.89 % without any augmentation. Through augmentation, its performance improves further to 99.92 %, slightly surpassing MobileNetV2's 99.87 %. The model's loss dramatically drops with augmentation, from 0.0054 to 0.0032, indicating that it can handle diverse datasets effectively. Additionally, RTR_Lite_MobileNet's precision, recall, and F1-score values are constantly above 99.85 %, indicating its ability to maintain balanced predictions across all classes. The suggested model also shows a 53.80 % reduction in parameters, from 2273,831 in MobileNetV2 to 1050,547, with no loss of performance. Performance trends further highlight RTR_Lite_MobileNet's robustness across both scenarios, with consistently lower loss and higher accuracy across epochs, as shown in the training metrics [Fig. 9\(a\)](#). On the Coffee Dataset [53], RTR_Lite_MobileNet outperforms MobileNetV2 by 1.14 %, achieving 89.00 % accuracy without augmentation. With augmentation, the accuracy increases to 90.84 %, 2.29 % higher than MobileNetV2's 88.55 %. The model demonstrates strong performance under enhanced settings, with precision (90.26 %), recall (90.32 %), and F1-score (90.23 %). The drop in loss from 0.4007 to 0.3858 indicate improved generalization skills of the model. The benefits of augmentation are clearly observed in higher validation accuracy and lower loss as shown in [Fig. 9\(b\)](#).

Without augmentation, RTR_Lite_MobileNet achieves an excellent accuracy of 97.62 % on the Wheat Dataset [54], which is comparable to MobileNetV2. However, augmentation provides significant benefits, as RTR_Lite_MobileNet reaches a remarkable 100 % accuracy, outperforming MobileNetV2 by 1.59 %. The model's remarkable ability to correctly identify all categories is demonstrated by its 100 % F1-score, precision, and recall scores. [Fig. 9\(c\)](#) highlights the rapid convergence of RTR_Lite_MobileNet on this dataset, with validation accuracy reaching 100 % in fewer epochs.

On Soybean dataset [55], without augmentation, RTR_Lite_MobileNet achieve accuracy of 94.07 % that is 2.03 % higher than MobileNetV2 (92.04 %). Its accuracy increases to 96.78 % with augmentation, outperforming MobileNetV2's 95.88 %. Metrics such as F1-score (96.75 %), recall (96.73 %), and precision (96.78 %) demonstrate how well it handles imbalanced classes. The loss reduction from 0.2834 to 0.1243 indicates increased confidence in predictions. As shown in [Fig. 9\(d\)](#), the validation accuracy and loss curves reflect RTR_Lite_MobileNet's improved learning trajectory with augmentation, ensuring both accuracy and stability.

The Sugarcane Dataset [56] shows that RTR_Lite_MobileNet outperforms MobileNetV2 with 96.67 % accuracy without augmentation. This highlights the model's ability to perform consistently across small,

balanced datasets. Metrics like recall (96.67 %), precision (94.44 %), and F1-score (95.22 %) demonstrate its effective prediction balance. [Fig. 9\(e\)](#) illustrates the model's rapid convergence and consistently low loss values.

The PlantDoc dataset [57] presents a unique challenge as it includes a variety of real-world photos. RTR_Lite_MobileNet achieves 64.73 % accuracy without augmentation, 5.69 % higher than MobileNetV2's 61.24 %. With augmentation, accuracy increases dramatically to 82.00 %, compared to MobileNetV2's 78.48 %. The loss reduction from 1.1796 to 0.6416 demonstrates its capacity to handle variations such as lighting and occlusion. [Fig. 9\(f\)](#) shows the significant impact of augmentation, as both accuracy and loss metrics improve substantially with RTR_Lite_MobileNet outperforming MobileNetV2 across all epochs. On the PaddyDoctor dataset [58], RTR_Lite_MobileNet surpasses MobileNetV2, achieving an accuracy of 96.92 % without augmentation. RTR_Lite_MobileNet with augmentation achieves an accuracy rate of 97.11 %, illustrating its exceptional precision. With the model's loss reduced to 0.1110, the reliability of predictions. The model's strength is evidenced by metrics such as F1-score (96.42 %), recall (96.61 %), and precision (96.48 %). Despite a 53.23 % reduction in parameters, RTR_Lite_MobileNet retains its computational efficiency. [Fig. 9\(g\)](#) highlights how the model achieves stable and consistent performance across epochs, with augmentation further enhancing accuracy and reducing loss.

[Figs. 10 -11](#) present a comparative analysis of accuracy and loss across different models on the datasets, emphasizing performance variations and the significant impact of data augmentation on improving model outcomes. The findings of the confusion matrix generated during the testing of the suggested model on multiple datasets are shown in [Fig. 12](#). A thorough evaluation of the effectiveness of the model across various classes in each dataset is given by the confusion matrix.

[Fig. 12](#) shows the integrated confusion matrices and ROC curves for the RTR_Lite_MobileNet model's performance when tested on the Plant Disease Dataset, PlantDoc, and PaddyDoctor. Similarly, [Fig. 13](#) presents the confusion matrices and ROC curves for the RTR_Lite_MobileNet model when evaluated on the test datasets of Coffee, Wheat, Soybean, and Sugarcane.

By highlighting true positives along the diagonal and presenting misclassifications in off-diagonal regions, confusion matrices provide a clear visual illustration of the model's ability to accurately forecast each class. ROC curves offer an in-depth assessment of a model's classification capabilities by illustrating the relationship between the true positive rate (sensitivity) and the false positive rate across all classes. The model demonstrated exceptional capability in handling complex disease patterns, achieving nearly flawless classification across all 39 categories and minimal misclassifications for the Plant Disease dataset. In the same spirit, the model exhibited a notable enhancement in performance on the PlantDoc dataset following augmentation, showcasing its adaptability in managing challenging real-world datasets.

The model's performance on PaddyDoctor dataset was impressive, especially for diseases like "blast" class index 3 and "brown spot," class index 4 where it was perfect. But there were a few minor misclassifications for visually similar categories, such as "bacterial_leaf_blight" index 1 and "bacterial_panicle_blight" Index 2.

Due to overlapping visual cues, the model had considerable difficulty differentiating between "Rust" index 0 and "Red Spider Mite," index 2 but it showed high precision and recall for "healthy" leaves on the Coffee dataset.

Similarly, datasets including Wheat, and Soybean showed high overall accuracy, with just slight performance differences in categories with mildly different symptoms. Finally, the confusion matrix supports accurate classification across its three classes in the balanced Sugarcane Dataset, which was not augmented, and the ROC curve shows consistently high AUC values.

Grad-CAM serves as an effective visualization tool that aids in interpreting and comprehending the decisions made by deep learning

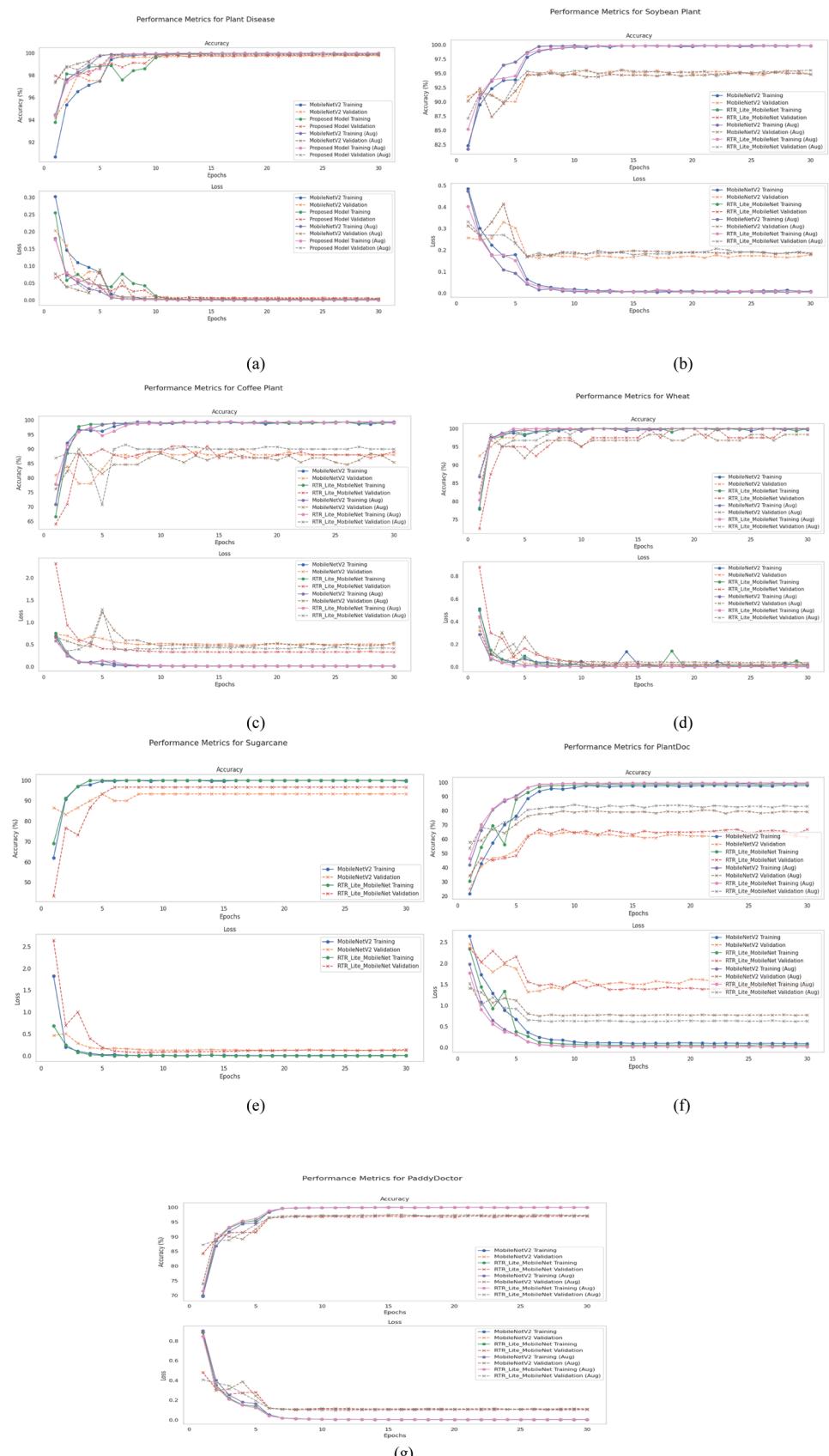


Fig. 9. Illustration of the training and validation accuracy achieved by the Standard MobileNetV2 and RTR_Lite_MobileNet models, both without and with augmentation, evaluated across seven diverse datasets: (a) Plant Disease Dataset [52]; (b) Soybean Plant [53]; (c) Coffee Plant [54]; (d) Wheat [55]; (e) Sugarcane [56]; (f) PlantDoc Dataset [57]; (g) PaddyDoctor Dataset [58].

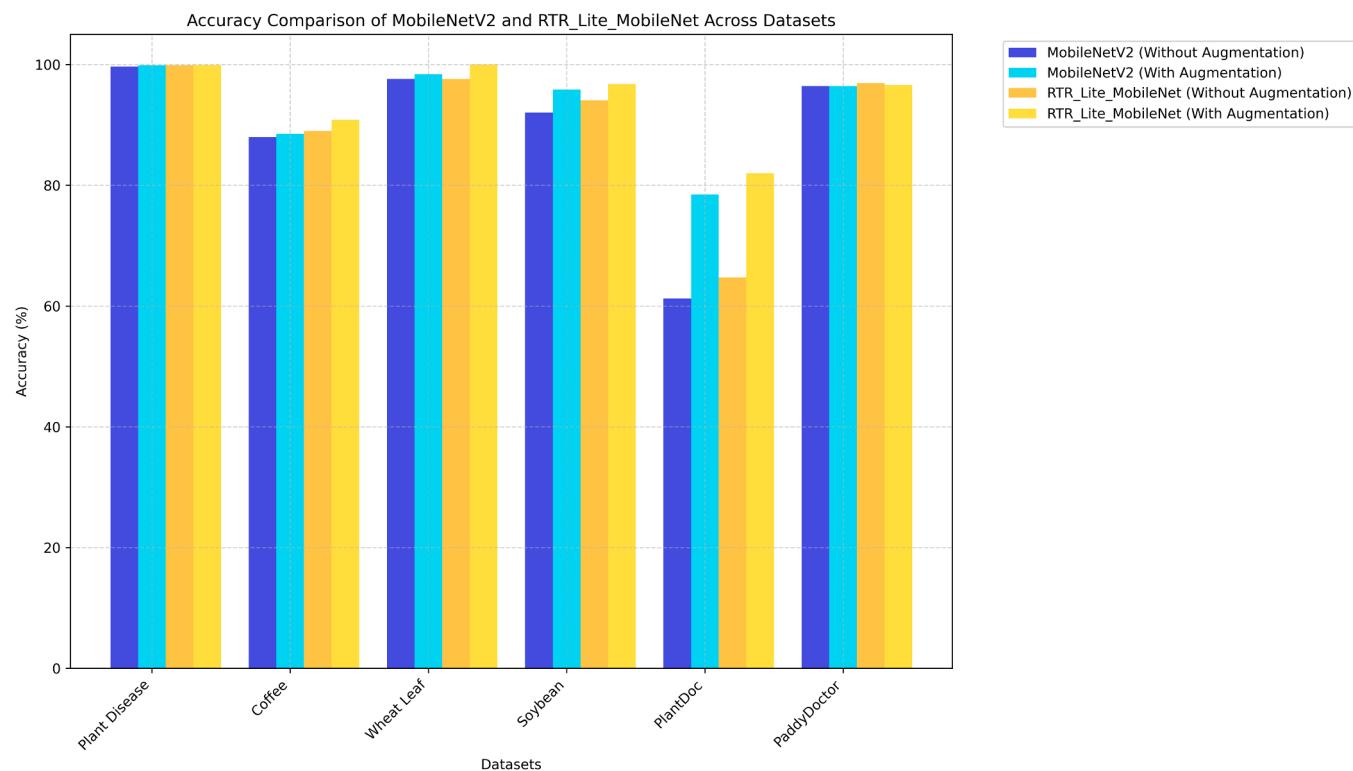


Fig. 10. Comparative analysis of the accuracy of different models on the datasets, emphasizing performance variations and the significant impact of data augmentation on improving model outcomes.

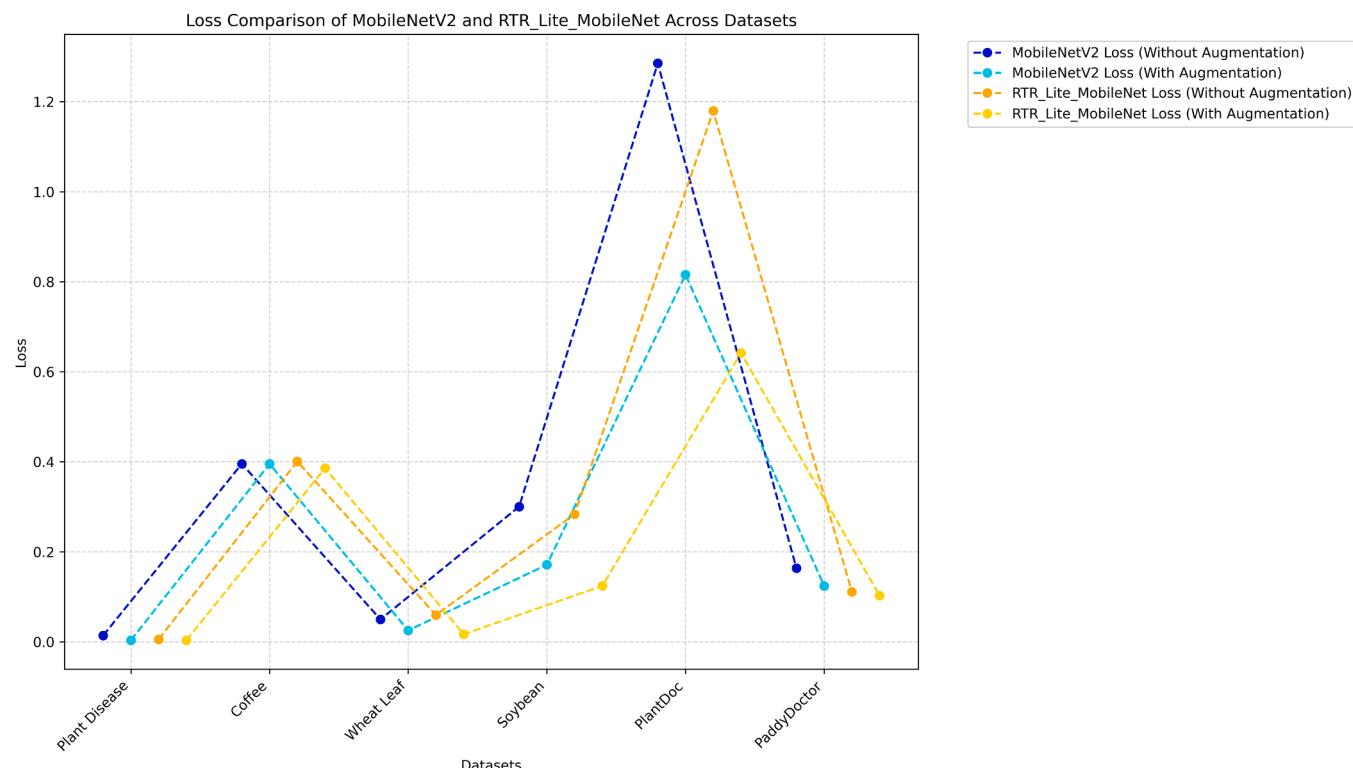
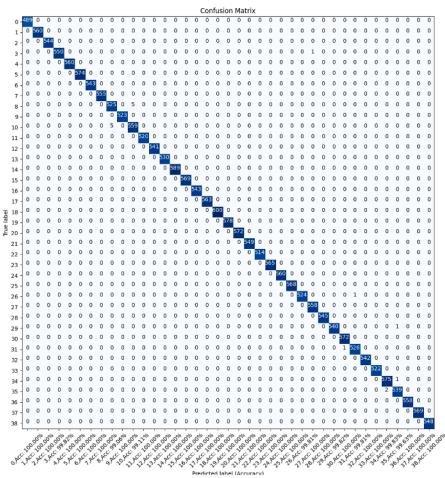


Fig. 11. Comparative analysis of loss across different models on the datasets, emphasizing performance variations and the significant impact of data augmentation on improving model outcomes.

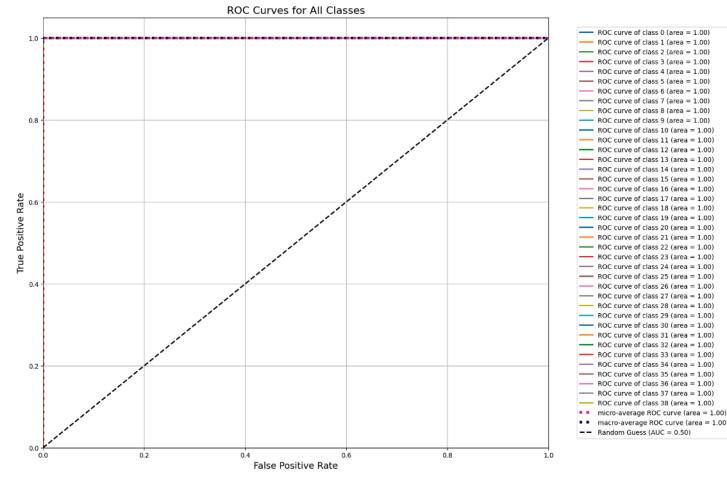
models. Through the creation of heatmaps, Grad-CAM emphasizes the areas in an input image that have a substantial impact on the model's predictions, facilitating the detection of essential features like disease

spots on plant leaves. This method not only helps confirm the dependability of the model's predictions but also offers understanding into the model's areas of emphasis. The results presented in Fig. 14 demonstrate

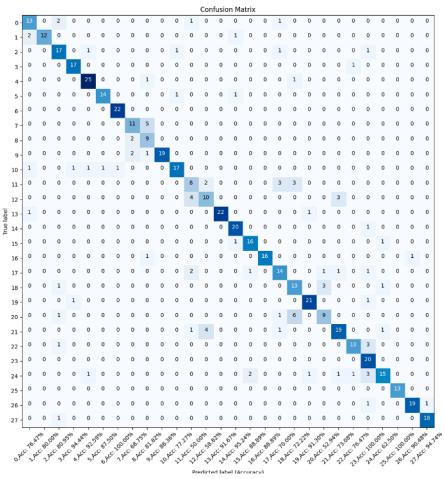
Plant Disease Dataset - Confusion Matrix



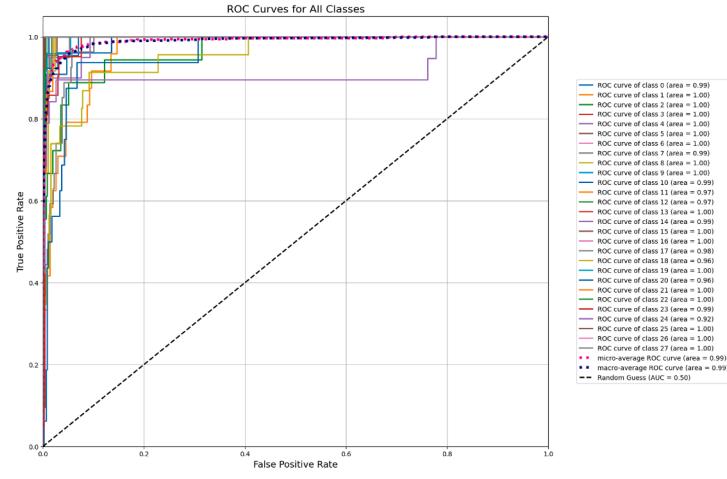
Plant Disease Dataset - ROC Curve



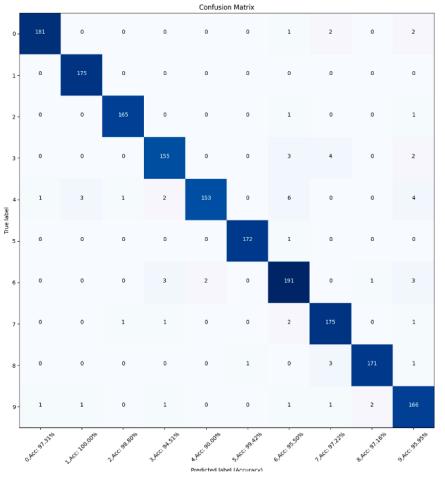
PlantDoc - Confusion Matrix



PlantDoc - ROC Curve



PaddyDoctor - Confusion Matrix



PaddyDoctor - ROC Curve

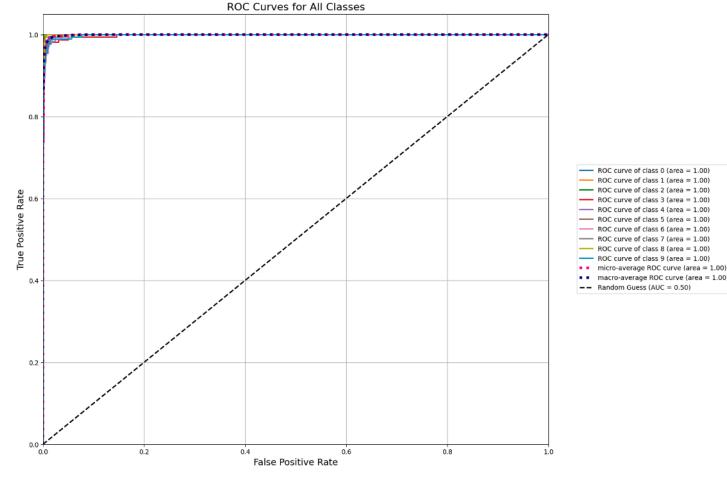


Fig. 12. Confusion Matrices and ROC Curves Illustrating Testing Performance of the RTR_Lite_MobileNet Model on Augmented Datasets from Plant Disease [52], PlantDoc [57], and PaddyDoctor [58].

the use of Grad-CAM to visualize the performance of the proposed RTR_Lite_MobileNet model in comparison to the MobileNetV2 model across various datasets.

3.6. Comparative analysis with other proposed lightweight models

In this section, a comparative analysis is conducted on various

lightweight models proposed for plant disease identification in recent studies. For this analysis, we specifically consider studies that proposed lightweight models and evaluated them using the PlantVillage dataset, referred to in this study as the Plant Disease Dataset [48]. Table 4 provides an overview of the comparative analysis of related lightweight models and our proposed model. Raj Kumar et al. [23] implemented a CNN approach, achieving 98.20 % accuracy across 8 classes in the

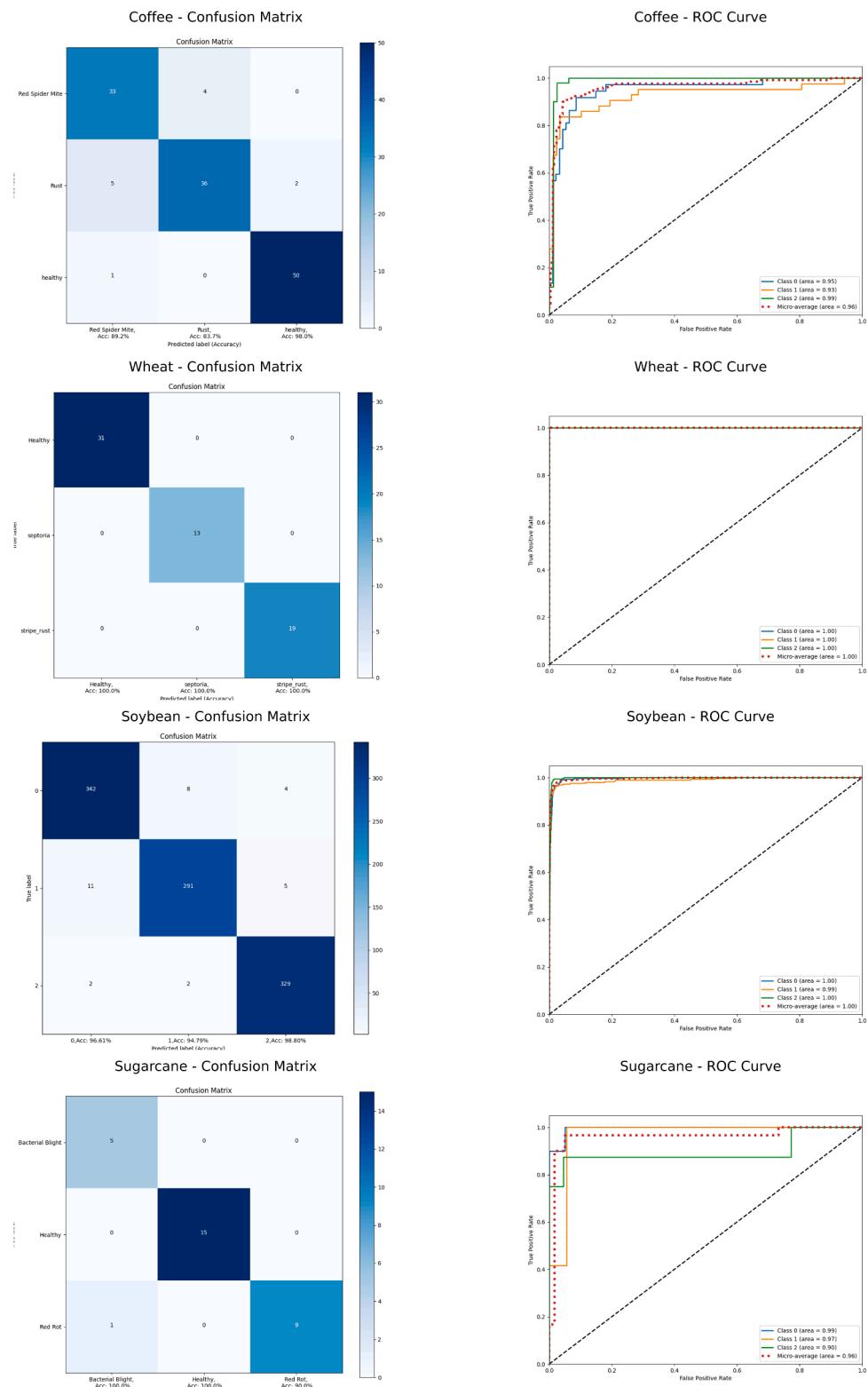


Fig. 13. Confusion Matrices and ROC Curves Illustrating Testing Performance of the RTR_Lite_MobileNet Model on Augmented Datasets from Coffee [53], Wheat [54], Soybean [55], Sugarcane [56].

PlantVillage dataset. Sutaji and Yildiz [25] utilized the LEMOXINET model (MobileNetV2 + Xception), achieving 99.10 % accuracy over 39 classes. Altalak et al. [26] integrated ResNet101 with CBAM and SVM, achieving 97.2 % accuracy across 10 classes. Biswas et al. [12] used CNN that classifying 38 classes with 95.17 % accuracy. Chen et al. [28] combined MobileNetv2, DenseNet, and SCAM, achieving 98.50 %

accuracy across 4 classes. Shafik et al. [30] proposed PDDNet-AE and PDDNet-LVE, achieving 96.74 % and 97.79 % accuracy respectively, across 39 classes. Rakib et al. [32] employed a quantized CNN, achieving 98 % accuracy across 9 classes, while R. Maurya et al. [33] employed MLP, LSTM, and SVM achieving 97.45 % accuracy over 15 classes. Although studies by Sutaji et al. [25] and Dheeraj et al. [34]

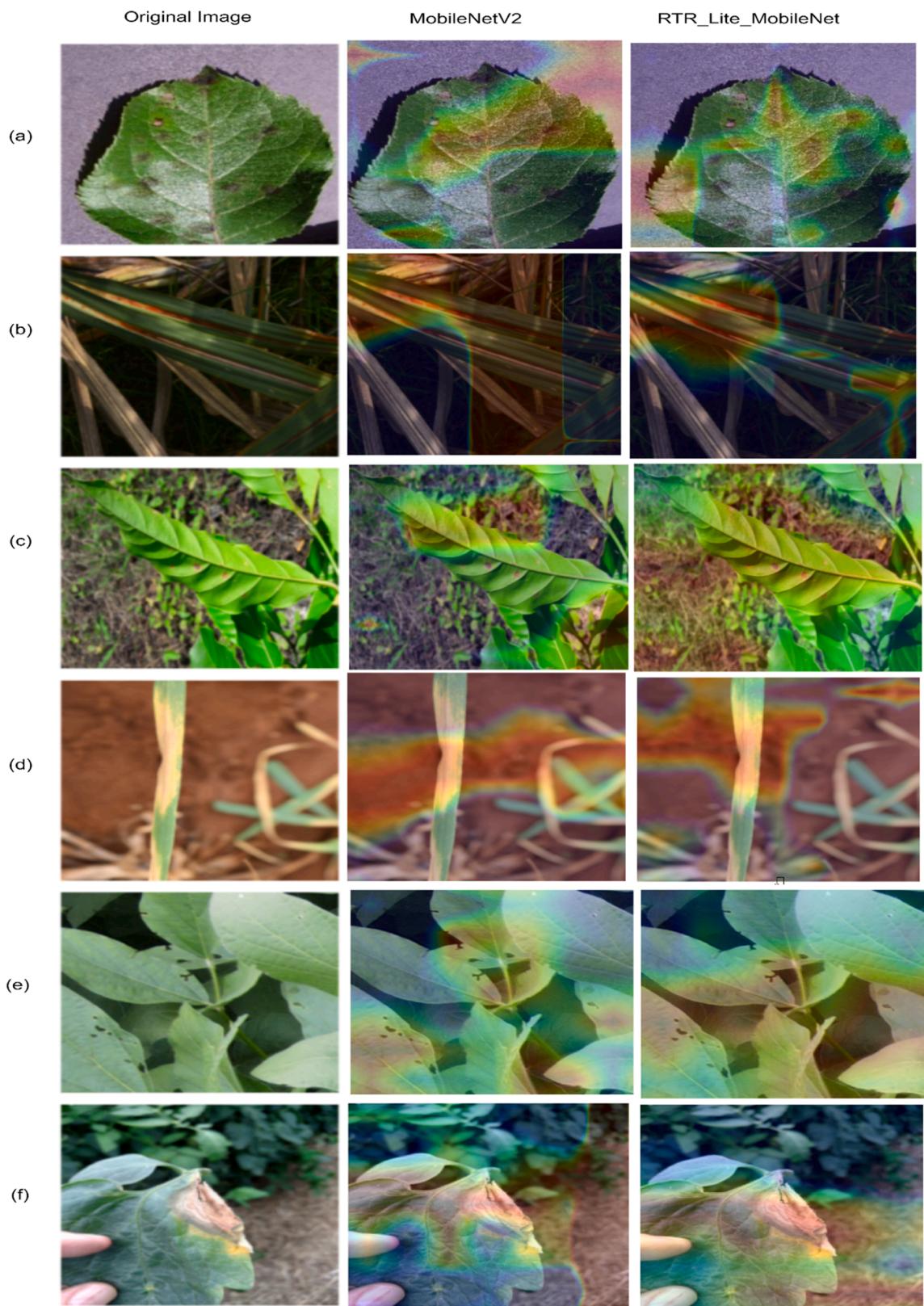


Fig. 14. GradCAMs visualization results of MobileNetV2 and RTR_Lite_MobileNet for datasets: (a) Plant Disease Dataset [52]; (b) Soybean Plant [53]; (c) Coffee Plant [54]; (d) Wheat [55]; (e) Sugarcane [56]; (f) PlantDoc Dataset [57]; (g) PaddyDoctor.Dataset [58].

Table 4
Related work for comparison analysis.

References	Methodology	Dataset	Accuracy
Raj kumar et al. [23]	CNN	PlantVillage – 8 classes	98.20 %
Sutaji and Yildiz [25]	LEMOXINET (MobileNetV2 + Xception) model	Plantvillage – 39 classes	99.10 %
Altalak et al. [26]	ResNet101 + CBAM + SVM	PlantVillage – 10 classes	97.2 %
Biswas et al. [12]	CNN	PlantVillage – 38 classes	95.17
Chen et al. [28]	Mobilenetv2 + DenseNet + SCAM	PlantVillage – 4 classes	98.50 %
Shafik et al. [30]	PDDNet-AE (Early Fusion), PDDNet-LVE (Major voting)	PlantVillage – 39 classes	96.74 % - PDDNet-AE 97.79 % - PDDNet-LVE
Rakib et al. [32]	Quantized CNN	PlantVillage - 9 classes	98 %
R. Maurya et al. [33]	MLP + LSTM + SVM	Plantvillage – 15 classes	97.45 %
Dheeraj et al. [34]	DenseNet121	PlantVillage – 39 classes	99.39 %
Assaduzzaman et al. [35]	EfficientNetB0 + SE module	PlantVillage – 11 classes	99.11 %
Thai & Le, [36]	CNN + Transformer	PlantVillage – 39 classes	99.27 %
Proposed model	RTR_Lite_MobileNet	PlantVillage - 39 classes	99.89 % (without augmentation) 99.92 % (with augmentation)

achieved high accuracy, their parameter counts remain high, approximately 26.57 million and 1.5 million respectively, which is higher than our proposed model's parameter count of 1.05 million. Assaduzzaman et al. [35], proposed the lightweight XSE-TomatoNet model (EfficientNetB0 + SE module), achieving 99.11 % accuracy on the PlantVillage dataset (11 classes), while in study [36], CNN and Transformer were utilized that achieves 99.27 % accuracy on the PlantVillage dataset (39 classes). The proposed RTR_Lite_MobileNet model achieved the highest accuracy of 99.89 % across 39 classes without augmentation and 99.92 % with augmentation, underscoring its effectiveness in plant disease classification using the PlantVillage dataset.

4. Ablation experiments

To determine the most effective optimizer for the proposed model, experiments were conducted using four different optimizers: ADAM, ADAMW, SGD, and RMSProp during the training phase on the Plant Disease dataset without augmentation. The performance of the RTR_Lite_MobileNet model was evaluated, and the results are presented in Table 5. Analyzing the data in Table 5, it is evident that the proposed model achieved testing accuracies of 99.89 %, 99.84 %, 92.02 %, and 99.77 % for the Adam, AdamW, SGD, and RMSProp optimizers, respectively. Notably, the highest accuracy was obtained with the Adam optimizer. Thus, for both training and testing phases of the proposed model, Adam optimizer was employed. After the optimizer is selected, proposed model is evaluated considering different learning rates, and execution results are presented in Table 6.

Table 5
Performance comparison of RTR_Lite_MobileNet using different optimizers.

Model	Model Size	Total parameters	Loss	Accuracy	Precision	Recall	F1-score	AUC- Score	Kappa Score
Adam	4.01	1050,547	0.0054	99.89 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %
AdamW			0.0085	99.84 %	99.81 %	99.75 %	99.78 %	100.00 %	99.83 %
SGD			0.2921	90.75 %	90.04 %	90.25 %	99.78 %	91.68 %	90.75 %
RMSprop			0.0127	99.77 %	99.75 %	99.71 %	99.73 %	99.99 %	99.76 %

For evaluating the impact of transfer learning (TL) on the proposed model, we utilized the MobileNetV2 model that had been pre-trained on the IMAGENET1K_V1 dataset. Experiment results are shown in Table 7. When analyzing RTR_Lite_MobileNet with and without transfer learning, it became evident that there were notable enhancements in performance metrics. The model utilizing transfer learning achieved a significantly lower loss of 0.0054 and a remarkable accuracy of 99.89 %. RTR_Lite_MobileNet without transfer learning has less numbers of parameters, hence the inference time of the models is lower than that of the pretrained RTR_Lite_MobileNet.

4.1. Impact of attention mechanisms and architectural modifications

Table 8 displays the outcomes of ablation investigations performed on the suggested model, MobileNetV2, with various modifications, and Fig. 15 represent the GradCAM saliency map visualization results of various models.

Model 1: The development of Model 1 involved the introduction of the triplet attention module, replacing the initial feature layer of the original MobileNetV2. In model 1, the last three layers of the original MobileNetV2 are dropped. It has a model size of 3.91MB, with 1024,271 total parameters, of which 697,792 are trainable parameters and 326,479 are non-trainable parameters. The model achieves a testing accuracy of 99.74 %.

Model 2: To test the effect of the ECA module, we replaced the triplet attention module in Model 1 with the ECA module. The total parameter count of Model 2 is 1024,075, of which 697,792 are trainable parameters and 326,283 are non-trainable parameters. Model 2 achieves a testing accuracy of 99.28 %, which is lower compared to Model 1.

Model 3: Model 3 is similar to Model 2, but instead of just adding the ECA module, we added RES_BLOCK, which contains SENet and the ECA module. Model 3 has a total of 1024,331 parameters, of which 697,792 are trainable and 326,539 are non-trainable parameters. Model 3 achieves a testing accuracy of 99.67 %, which is higher than Model 2 but lower than Model 1.

Model 4: In Model 4, instead of just adding one attention block, both triplet attention and RES_Block are added in the initial layer of MobileNetV2 for efficient extraction of relevant features. Now, the model size remains the same, but now the total parameters are 1,024,531. After the introduction of RES_Block, the model achieves a testing accuracy of 99.84 %, which is a 0.10 % increment compared to Model 1.

Model 5: In Model 3, the initial features remain the same as in Model 2, but the last layers of MobileNetV2, which were dropped, are replaced with RES_BLOCK, TAM, and RES_BLOCK, respectively. Model 5 has a model size of 3.96 and a total parameter count of 1,037,539, of which 1,024,396 are trainable and 13,143 are non-trainable parameters. It achieves a testing accuracy of 99.79 %. The addition of attention blocks increases the model complexity.

Model 6 (proposed model): Model 6 is modified. Model 5, in which the same architecture is followed, but to increase the model's performance in identifying the disease-related regions in the image. To achieve this, the last RES_BLOCK, TAM, and RES_BLOCK are repeated one more time just before the classifier layer. By doing this, the model size becomes 4.01 MB, and the total parameter counts is 1,050,547, of which 1,037,404 are trainable parameters and 13,143 are non-trainable parameters. The model achieves a high testing accuracy of 99.89 %. As evident from Fig. 15 (g), the model correctly identify each disease spot

Table 6

Performance comparison of RTR_Lite_MobileNet using different learning rate.

Model	Learning Rate	Loss	Accuracy	Precision	Recall	F1-score	AUC- Score	Kappa Score
RTR_Lite_MobileNet	0.001	0.0054	99.89 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %
	0.003	0.0145	99.74 %	99.66 %	99.68 %	99.67 %	99.99 %	99.73 %
	0.005	0.0118	99.67 %	99.60 %	99.56 %	99.57 %	99.99 %	99.66 %

Table 7

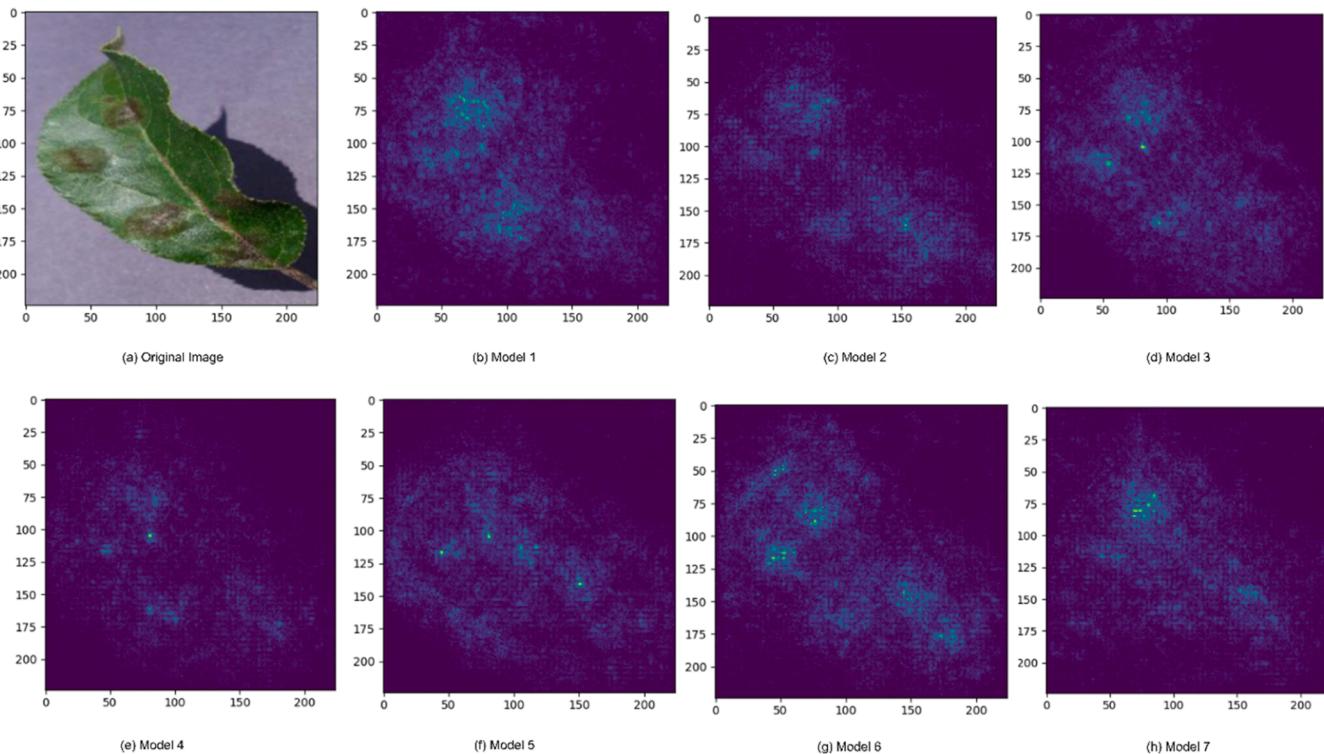
Performance comparison of RTR_Lite_MobileNet with or without Transfer learning.

Model	Parameters count	Loss	Accuracy	Precision	Recall	F1-score	AUC- Score	Kappa Score
RTR_Lite_MobileNet with TL	1,050,547	0.0054	99.8 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %
RTR_Lite_MobileNet without TL	1,024,691	0.0190	99.4 %	99.23 %	99.28 %	99.25 %	100.00 %	99.41 %

Table 8

Results of ablation experiments on proposed model.

Model	Model Size	Total parameters	Loss	Accuracy	Precision	Recall	F1-score	AUC- Score	Kappa Score
Model 1	3.91	1,024,271	0.0090	99.74 %	99.65 %	99.66 %	99.66 %	99.99 %	99.73 %
Model 2	3.91	1,024,075	0.0233	99.27 %	99.09 %	99.07 %	99.09 %	100.00 %	99.27 %
Model 3	3.91	1,024,331	0.0103	99.67 %	99.58 %	99.59 %	99.58 %	100.00 %	99.66 %
Model 4	3.91	1,024,531	0.0068	99.84 %	99.76 %	99.79 %	99.77 %	100.00 %	99.83 %
Model 5	3.96	1,037,799	0.0142	99.79 %	99.74 %	99.70 %	99.72 %	100.00 %	99.78 %
Model 6	4.01	1,050,547	0.0054	99.89 %	99.87 %	99.85 %	99.86 %	100.00 %	99.88 %
Model 7	3.91	1,024,691	0.0106	99.74 %	99.66 %	99.66 %	99.66 %	100.00 %	99.73 %

**Fig. 15.** GradCAM Saliency Map results of the ablation experiment on RTR_Lite_MobileNet: (a) Original image; (b) Model 1; (c) Model 2; (d) Model 3; (e) Model 4; (f) Model 5; (g) Model 6; (h) Model 7.

on the given image, showcasing its capability to capture more meaningful and focused regions of the image, validating the enhanced feature representation.

Model 7: Model 7 is created by modifying Model 6, as already discussed in the section above. With the addition of the attention block in the end, the model's inference time increases. To tackle this, in this model instead of RES_Block, only the ECA module is used due to its

lightweight nature. For this, the TAM and ECA modules are first introduced in the initial layer, and after removing the last three layers of the original MobileNetV2, the ECA, TAM, and ECA modules are added, and repeated twice as done in Model 4. The Model 7 has 1,024,691 total parameters, of which 1,018,204 are trainable and 6487 are non-trainable. Model 7 achieves a testing accuracy of 99.74 %. As shown in Fig. 13 (h), Model 7 cannot identify all disease areas in the image like

Model 6 (the suggested model). These results provide insights into how different modifications to the MobileNetV2 architecture and choice of optimizer impact its performance.

5. Performance evaluation on IoT devices

In this study, the widely accepted PyTorch deep learning framework was used to create and train the proposed model for plant disease classification. PyTorch was chosen primarily because of its adaptability in both model development and training. For testing the proposed model's efficacy on IoT devices, the model is converted to TensorFlow Lite; the whole procedure carried out is depicted in Fig. 16. The model was transferred to the ONNX (Open Neural Network Exchange) format after training was finished.

ONNX functions as a bridge representation, facilitating compatibility across frameworks, enhancing translation efficiency, and minimizing dependencies on specific frameworks. The ONNX model was then converted to TensorFlow Lite, a format appropriate for deployment on resource-constrained edge devices. This conversion phase was essential for enhancing the model's performance by decreasing computational complexity and memory usage. The TensorFlow Lite model was ultimately implemented on multiple edge devices, including Raspberry Pi 4, Raspberry Pi 5, and Jetson Nvidia Nano, where inferences speed, latency, and compute performance were evaluated.

Prior to deploying the RTR_Lite_MobileNet and MobileNetV2 models on IoT devices like Raspberry Pi 4 and Raspberry Pi 5, quantization is performed on TensorFlow Lite models to enhance performance. Table 9 shows the results of both models' deployments. The findings indicate that RTR_Lite_MobileNet outperforms MobileNetV2 in terms of overall performance, with much lower CPU and GPU latency and RAM utilization. Despite these improvements, latency reduction could yet be improved. To solve this, future research can look into methods like structured pruning, lightweight attention mechanisms, hardware-specific optimizations, and dynamic resolution adaption. These methods have the potential to further reduce computational demands and latency.

6. Discussion and future scope

The proposed RTR_Lite_MobileNet integrates advanced attention mechanism like Triplet Attention, SENet, and the ECA module to improve the model's disease-related feature extraction ability. The study found that the RTR_Lite_MobileNet model outperforms MobileNetV2 across the Plant Disease, Coffee, PlantDoc, Soybean, Sugarcane, Wheat, and PaddyDoctor datasets. Without augmentation, the model achieved 99.89 % accuracy on the Plant Disease dataset, surpassing MobileNetV2 by 0.23 %. Augmentation further increased accuracy to 99.92 %, while reducing loss from 0.0054 to 0.0032. The model's 82.00 % accuracy on the difficult PlantDoc dataset shows it can handle complex real-world photos. The model's 97.11 % accuracy on the PaddyDoctor dataset proved its reliability, while its loss was minimized to 0.1110. Similarly, RTR_Lite_MobileNet achieved 90.84 % accuracy on the Coffee dataset. For the Wheat dataset, the model achieved 100 % classification accuracy with augmentation, surpassing MobileNetV2 by 1.59 %. The Soybean dataset showed a similar pattern, with the model achieving 94.07 % accuracy without augmentation and improving to 96.78 % with

augmentation, reducing loss from 0.2834 to 0.1243. The Sugarcane dataset, which is balanced and does not require augmentation, achieved 96.67 % accuracy, demonstrating the model's inherent robustness and efficiency in smaller datasets.

Deployment findings on IoT devices supported the model's efficiency. On Raspberry Pi 4, RTR_Lite_MobileNet achieved a CPU latency of 78 ms compared to MobileNetV2's 81 ms (a 3.85 % improvement) and decreased GPU latency to 13 ms versus 14 ms (7.69 % improvement). On Raspberry Pi 5, the model had a CPU latency of 136 ms, compared to MobileNetV2's 142 ms (4.41 % improvement), and a GPU delay of 22 ms versus 24 ms (a 9.09 % improvement). These results highlight the model's computational efficiency, making it highly suitable for real-time agricultural applications. The proposed model is efficient for use in large and complex datasets as well as for multi-classification tasks.

While RTR_Lite_MobileNet demonstrates significant improvements in accuracy, efficiency, and adaptability for plant disease classification, it is important to acknowledge certain limitations. Crop disease identification exhibits inherent variability influenced by factors such as seasonality, crop maturity, foliar damage, and environmental conditions, including light and background complexity. These parameters can significantly impact the model's performance; nevertheless, they are inadequately examined in the present work. Future efforts will focus on conducting field trials to assess the model's robustness under real-world settings. This evaluation will provide critical insights into its dependability and suitability for actual agricultural monitoring. Furthermore, the datasets used in this study could not accurately reflect the variation seen in real-world agricultural environments. By simulating realistic variations in disease patterns, light, and leaf conditions using synthetic data generation techniques—like GANs—data diversity will be improved. This approach will improve the capacity of our model to generalize over several contexts. External dataset cross-validation will assess the model's adaptability to new data and improve its global agricultural reliability. RTR_Lite_MobileNet uses attention techniques that add computational complexity and somewhat slow down inference, despite their advantages. The suggested architecture still has a higher FLOP count than MobileNetV3 and ShuffleNetV2, necessitating further optimization. Future investigations will delve into intricate techniques such as quantization-aware training and structured pruning to tackle these challenges. In order to reduce computational expenses and enhance the model's scalability and adaptability for deployment across various IoT devices, an examination of lightweight attention techniques will be conducted. Additionally, to augment datasets with authentic variances and guarantee resilience across various agricultural contexts, forthcoming research will integrate synthetic data generation methods, such as GANs, to improve data diversity and generalizability. The cross-validation of external datasets will be prioritized to assess the model's adaptability to real-world conditions and data scarcity [62,63].

Furthermore, edge-cloud hybrid systems will be examined to ensure continuous connectivity between cloud platforms and IoT devices. This approach will facilitate efficient scaling for large-scale agricultural monitoring systems by offloading resource-intensive operations to the cloud, while maintaining real-time forecasts at the edge [64]. By further improving RTR_Lite_MobileNet, these developments hope to make it a more reliable, effective, and highly flexible solution for actual agricultural problems, especially in settings with limited resources where reliability and effectiveness are crucial.

7. Conclusion

This study introduces RTR_Lite_MobileNet, an improved and streamlined version of the MobileNetV2 model that incorporates several attention mechanisms, including triplet attention, SENet, and ECA. The suggested approach outperforms various standard classification models, including ResNet101, EfficientNetV2, ShuffleNetV2, ViT_B_16, and MobileNetV2, on the Plant Disease Dataset. RTR_Lite_MobileNet

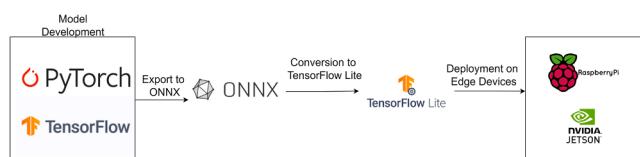


Fig. 16. Workflow for deploying RTR_Lite_MobileNet on Resource-Constrained IoT Devices.

Table 9

Performance Comparison of RTR_Lite_MobileNet and MobileNetV2 on IoT.

Model	Device	RAM usage (MB)	ROM usage (tflite model) (MB)	Latency (CPU)	Latency (GPU)
RTR_Lite_MobileNet	Raspberry Pi 4	2.5	1.7	78 ms	13 ms
MobileNetV2		2.6	2.5	81 ms (+3.85 %)	14 ms (+7.69 %)
RTR_Lite_MobileNet	Raspberry Pi 5	2.5	1.7	136 ms	22 ms
MobileNetV2		3.0	2.5	142 ms (+4.41 %)	24 ms (+9.09 %)

surpasses the MobileNetV2 model across seven publicly accessible datasets, demonstrating superior performance in many evaluation metrics like accuracy, precision, recall, and F1 scores, while also preserving a reduced model size and fewer trainable parameters. On the Plant Disease dataset, the RTR_Lite_MobileNet attained an accuracy of 99.89 %, which increased to 99.92 % with augmentation.

On the challenging PlantDoc dataset, the model achieved a classification accuracy of 82.00 % with augmentation, demonstrating its efficiency in detecting complicated patterns in real-world data. The findings of this study demonstrate that attention mechanisms can be efficiently used in neural networks to improve model performance. Alternative implementations of attention mechanisms could replace the current layer of the model, leading to a reduction in model size and parameter count. It is essential to conduct experiments to assess the comparability of the utilized attention mechanisms and deep learning models. The incorporation of advanced attention mechanisms markedly diminished the model's size and trainable parameters, hence enhancing its applicability for deployment on resource-limited IoT devices. This paper rigorously evaluates the proposed RTR_Lite_MobileNet model, validating its effectiveness and robustness for real-time plant disease detection through deployment on Raspberry Pi 4 and Raspberry Pi 5, where it achieved lower latency compared to MobileNetV2, demonstrating its practical applicability for real-time agricultural monitoring. Future research will focus on optimization techniques, broader dataset validation, and advanced methods to enhance the model's adaptability and efficiency in diverse scenarios.

Funding

The authors declare that they did not receive any funding or grants during the preparation of this manuscript.

CRediT authorship contribution statement

Sangeeta Duhan: Conceptualization, Data curation, Methodology, Writing – original draft, Preeti Gulia: Conceptualization, Validation, Review & Editing, Supervision, Nasib Singh Gill: Formal analysis, Validation, Review & Editing, Supervision, and Ekta Narwal: Validation, Review & Editing. All authors read and approved the final manuscript for publication.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request. The datasets for this study are available at the following sources: Plant Disease Dataset (<https://data.mendeley.com/datasets/tywbtsjrv/1>), Soybean Plant Dataset (<https://data.mendeley.com/datasets/bycbh73438/1>), Coffee Plant Disease Dataset (<https://data.mendeley.com/datasets/c5yvn32dzg/2>), Wheat Leaf Dataset (<https://www.kaggle.com/datasets/olyadgetch/wheat-leaf-dataset>), Sugarcane Disease Dataset (<https://www.kaggle.com/datasets/prabhakaransoundar/sugarcane-disease-dataset>), Plant-

Doc Dataset (<https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset>), and PaddyDoctor Dataset (<https://www.kaggle.com/datasets/imbikramsha/paddy-doctor>).

References

- [1] X. Jiang, Y. Sun, M. Shen, L. Tang, How does developing green agriculture affect poverty? Evidence from China's prefecture-level cities, Agric 14 (3) (2024) 402, <https://doi.org/10.3390/agriculture14030402>.
- [2] J. Rockström, J. Williams, G. Daily, A. Noble, N. Matthews, L. Gordon, H. Wetterstrand, F. DeClerck, M. Shah, P. Steduto, C. De Fraiture, N. Hatibu, O. Unver, J. Bird, L. Sibanda, J. Smith, Sustainable intensification of agriculture for human prosperity and global sustainability, Ambio 46 (1) (2016) 4–17, <https://doi.org/10.1007/s13280-016-0793-6>.
- [3] S. Savary, L. Willocquet, S.J. Pethybridge, P. Esker, N. McRoberts, A. Nelson, The global burden of pathogens and pests on major food crops, Nat. Ecol. Evol. 3 (3) (2019) 430–439, <https://doi.org/10.1038/s41559-018-0793-y>.
- [4] S. Yu, L. Xie, Q. Huang, Inception convolutional vision transformers for plant disease identification, Internet Things 21 (2023) 100650, <https://doi.org/10.1016/j.iot.2022.100650>.
- [5] T. Zhang, Y. Cai, P. Zhuang, J. Li, Remotely sensed crop disease monitoring by machine learning algorithms: a review, Unmanned Syst. 12 (1) (2024) 161–171, <https://doi.org/10.1142/S2301385024500237>.
- [6] A. Morchid, M. Marhoun, R. El Alami, et al., Intelligent detection for sustainable agriculture: a review of IoT-based embedded systems, cloud platforms, DL, and ML for plant disease detection, Multimed. Tools Appl. 83 (2024) 70961–71000, <https://doi.org/10.1007/s11042-024-1839-9>.
- [7] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)2017, pp. 2261–2269, <https://doi.org/10.1109/CVPR.2017.243>.
- [8] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–14, <https://doi.org/10.48550/arXiv.1409.1556>.
- [9] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778, <https://doi.org/10.1109/CVPR.2016.90>.
- [10] N. Sameera, B.A. Ram, P.S.S. Prasad, A. Lakshmanarao, Plant disease detection using MLP, Convnets and Densenet models, in: Proc. Fourth Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. (ICAECT), 2024, pp. 1–5, <https://doi.org/10.1109/ICAECT60202.2024.10468895>.
- [11] M. A. Rajab, F.A. Abdullatif, T. Sutikno, Classification of grapevine leaves images using VGG-16 and VGG-19 deep learning nets, Telkomnika 22 (2) (2024) 445, <https://doi.org/10.12928/telkomnika.v22i2.25840>.
- [12] S. Biswas, I. Saha, A. Deb, Plant disease identification using a novel time-effective CNN architecture, Multimed. Tools Appl. 83 (2024) 82199–82221, <https://doi.org/10.1007/s11042-024-1882-8>.
- [13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA2018, pp. 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>.
- [14] N. Ma, X. Zhang, H.T. Zheng, J. Sun, ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, ECCV 2018. Lecture Notes in Computer Science, 11218, Springer, Cham, 2018, pp. 615–630, https://doi.org/10.1007/978-3-030-01264-9_8.
- [15] M. Tan, Q. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, 2019, 97 pp.6105-6114(<https://proceedings.mlr.press/v97/tan19a.html>).
- [16] P.S. Thakur, S. Chaturvedi, P. Khanna, T. Sheorey, A. Ojha, Vision transformer meets convolutional neural network for plant disease classification, Ecol. Inform. 77 (2023) 102245, <https://doi.org/10.1016/j.ecoinf.2023.102245>.
- [17] F. Aghaei, E. Fazl-Ersi, H. Noori, MDSSD-MobV2: an embedded deconvolutional multispectral pedestrian detection based on SSD-MobileNetV2, Multimed. Tools Appl. (2023), <https://doi.org/10.1007/s11042-023-17188-7>.
- [18] S. Banarase, S. Shirbahadurkar, The orchard guard: deep learning powered apple leaf disease detection with MobileNetV2 model, 799, J. Integr. Sci. Technol. 12 (2024) 799, <https://doi.org/10.62110/sciencein.jist.2024.v12.799>.
- [19] R.K. Banoth, B.V.R. Murthy, Soil image classification using transfer learning approach: MobileNetV2 with CNN, SN Comput. Sci. 5 (1) (2024) 199, <https://doi.org/10.1007/s42979-023-02500-x>.

- [20] A. Arya, P.K. Mishra, MobileNetV2-Incep-M: a hybrid lightweight model for the classification of rice plant diseases, *Multimed. Tools Appl.* 83 (2024) 79117–79144, <https://doi.org/10.1007/s11042-024-18723-w>.
- [21] G. Liu, J. Peng, A.A.A. El-Latif, SK-MobileNet: a lightweight adaptive network based on complex deep transfer learning for plant disease recognition, *Arab. J. Sci. Eng.* 48 (2) (2023) 1661–1675, <https://doi.org/10.1007/s13369-022-06987-z>.
- [22] K.G. Panchbhai, M.G. Lanjewar, V.V. Malik, P. Charanarav, Small size CNN (CAS-CNN), and modified MobileNetV2 (CAS-MODMOBNET) to identify cashew nut and fruit diseases, *Multimed. Tools Appl.* 83 (2024) 89871–89891, <https://doi.org/10.1007/s11042-024-19042-w>.
- [23] R. Kumar, A. Chug, A.P. Singh, Plant foliage disease diagnosis using light-weight efficient sequential CNN model, *Opt. Mem. Neural Net.* 32 (4) (2023) 331–345, <https://doi.org/10.3103/S1060992X23040100>.
- [24] M. Alkanan, Y. Gulzar, Enhanced corn seed disease classification: Leveraging MobileNetV2 with feature augmentation and transfer learning, *Front. Appl. Math. Stat.* 9 (2024) 1320177, <https://doi.org/10.3389/fams.2023.1320177>.
- [25] D. Sutaji, O. Yıldız, LEMOXINET: Lite ensemble MobileNetV2 and Xception models to predict plant disease, *Ecol. Inform.* 70 (2022) 101698, <https://doi.org/10.1016/j.ecoinf.2022.101698>.
- [26] M. Altalak, M.A. Uddin, A. Alajmi, A. Rizg, A hybrid approach for the detection and classification of tomato leaf diseases, *Appl. Sci.* 12 (16) (2022) 8182, <https://doi.org/10.3390/app12168182>.
- [27] S. Woo, J. Park, J.Y. Lee, I.S. Kweon, CBAM: Convolutional Block Attention Module, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), *Computer Vision – ECCV 2018*, *ECCV 2018. Lecture Notes in Computer Science*, 11211, Springer, Cham, 2018, pp. 3–19, https://doi.org/10.1007/978-3-030-01234-2_1.
- [28] J. Chen, W. Wang, D. Zhang, A. Zeb, Y.A. Nanehkaran, Attention embedded lightweight network for maize disease recognition, *Plant Pathol.* 70 (3) (2020) 630–642, <https://doi.org/10.1111/ppa.13322>.
- [29] G. Geetharamani, A. Pandian, Identification of plant leaf diseases using a nine-layer deep convolutional neural network, *Comput. Electr. Eng.* 76 (2019) 323–338, <https://doi.org/10.1016/j.compeleceng.2019.04.011>.
- [30] W. Shafik, A. Tufail, C.D.S. Liyanage, R.A.A.H. Mohd Apong, Using transfer learning-based plant disease classification and detection for sustainable agriculture, *BMC Plant Biol.* 24 (1) (2024) 136, <https://doi.org/10.1186/s12870-024-04829-y>.
- [31] G. Dai, Z. Tian, J. Fan, C.K. Sunil, C. Dewi, DFN-PSAN: multi-level deep information feature fusion extraction network for interpretable plant disease classification, *Comput. Electron. Agric.* 216 (2024) 108481, <https://doi.org/10.1016/j.compag.2023.108481>.
- [32] A.F. Rakib, R. Rahman, A.A. Razi, A.S.M.T. Hasan, A lightweight quantized CNN model for plant disease recognition, *Arab. J. Sci. Eng.* 49 (3) (2023) 4097–4108, <https://doi.org/10.1007/s13369-023-08280-z>.
- [33] R. Maurya, S. Mahapatra, L. Rajput, A lightweight meta-ensemble approach for plant disease detection suitable for IOT-based environments, *IEEE Access* 12 (2024) 28096–28108, <https://doi.org/10.1109/access.2024.3367443>.
- [34] A. Dheeraj, S. Chand, LWDN: lightweight DenseNet model for plant disease diagnosis, *J. Plant Dis. Prot.* 131 (3) (2024) 1043–1059, <https://doi.org/10.1007/s41348-024-00915-z>.
- [35] M. Assaduzzaman, P. Bishshash, Md.A.S. Nirob, A.A. Marouf, J.G. Rokne, R. Alhajj, XSE-TomatoNet: an explainable AI based tomato leaf disease classification method using EfficientNetB0 with squeeze-and-excitation blocks and multi-scale feature fusion, *MethodsX* 14 (2025) 103159, <https://doi.org/10.1016/j.mex.2025.103159>.
- [36] H.-T. Thai, K.-H. Le, MobileH-Transformer: enabling real-time leaf disease detection using hybrid deep learning approach for smart agriculture, *Crop Prot.* 189 (2025) 107002, <https://doi.org/10.1016/j.cropro.2024.107002>.
- [37] M. Li, Z. Tao, W. Yan, S. Lin, K. Feng, Z. Zhang, Y. Jing, Apnet: Lightweight network for apricot tree disease and pest detection in real-world complex backgrounds, *Plant Methods* 21 (1) (2025) 4, <https://doi.org/10.1186/s13007-025-01324-5>.
- [38] K. Joshi, S. Hooda, A. Sharma, H. Sonah, R. Deshmukh, N. Tuteja, S.S. Gill, R. Gill, Precision diagnosis of tomato diseases for sustainable agriculture through deep learning approach with hybrid data augmentation, *Curr. Plant Biol.* 41 (2025) 100437, <https://doi.org/10.1016/j.cpb.2025.100437>.
- [39] D. Misra, T. Nalamada, A.U. Arasanipalai, Q. Hou, Rotate to attend: Convolutional triplet attention module, in: Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Waikoloa, HI, USA, 2021, pp. 3138–3147, <https://doi.org/10.1109/WACV48630.2021.00318>.
- [40] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7132–7141, <https://doi.org/10.1109/CVPR.2018.00745>.
- [41] Y. Li, X. Bai, C. Xia, An improved YOLOv5 based on triplet attention and prediction head optimization for marine organism detection on underwater mobile platforms, *J. Mar. Sci. Eng.* 10 (9) (2022) 1230, <https://doi.org/10.3390/jmse10091230>.
- [42] J. Ye, Z. Yuan, C. Qian, X. Li, CAA-YOLO: combined-attention-augmented YOLO for infrared ocean ships detection, *Sensors* 22 (10) (2022) 3782, <https://doi.org/10.3390/s22103782>.
- [43] P. Yuan, Y. Xia, Y. Tian, H. Xu, TRiP: a transfer learning-based rice disease phenotype recognition platform using SENet and microservices, *Front. Plant Sci.* 14 (2024) 1255015, <https://doi.org/10.3389/fpls.2023.1255015>.
- [44] S. Zhao, Y. Peng, J. Liu, S. Wu, Tomato leaf disease diagnosis based on improved convolution neural network by attention module, *Agriculture* 11 (7) (2021) 651, <https://doi.org/10.3390/agriculture11070651>.
- [45] Q. Zeng, L. Niu, S. Wang, W. Ni, SEViT: a large-scale and fine-grained plant disease classification model based on transformer and attention convolution, *Multimed. Syst.* 29 (2023) 1001–1010, <https://doi.org/10.1007/s00530-022-01034-1>.
- [46] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: efficient channel attention for deep convolutional neural networks, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, 2020, pp. 11531–11539, <https://doi.org/10.1109/CVPR42600.2020.01155>.
- [47] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM* 60 (6) (2017) 84–90, <https://doi.org/10.1145/3065386>.
- [48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 2818–2826, <https://doi.org/10.1109/CVPR.2016.308>.
- [49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proc. Comput. Vis.–ECCV 2016, in: *Lecture Notes Computer Science*, 9905, Springer, Cham, 2016, pp. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.
- [50] A. Howard, M. Sandler, B. Chen, W. Wang, L.C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, Q. Le, Searching for MobileNetV3, in: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)2019, pp. 1314–1324, <https://doi.org/10.1109/ICCV.2019.00140>.
- [51] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy2017, pp. 618–626, <https://doi.org/10.1109/ICCV.2017.74>.
- [52] A. Pandian, G. Gopal, Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network, Dataset, (2019) Retrieved December 15, 2024, from Mendeley Data: <https://doi.org/10.17632/tywbtjsrjv.1>.
- [53] Parraga-Alava, J., Cusme, K., Loor, A., & Santander, E. RoCoLe: A robust coffee leaf images dataset. (2019). Retrieved December 15, 2024, from Mendeley Data: <https://doi.org/10.17632/c5yvn32dzg.2>.
- [54] Olyad Getch, Wheat Leaf dataset, (2021). Retrieved December 15, 2024, from Kaggle website: (<https://www.kaggle.com/datasets/olyadgetch/wheat-leaf-data-set>).
- [55] M.E. Mignoni, Images of soybean leaves, (2021). Retrieved from Mendeley Data: <https://doi.org/10.17632/bycbh73438.1>.
- [56] P. Soundar, Sugarcane Disease Dataset. (2022). Retrieved December 15, 2024, from Kaggle website: (<https://www.kaggle.com/datasets/prabhakaransoundar/sugarcane-disease-dataset>).
- [57] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, N. Batra, PlantDoc: a dataset for visual plant disease detection (New York, NY, USA: Association for Computing Machinery), Proc. 7th ACM IKDD CoDS 25th COMAD (2020) 249–253, <https://doi.org/10.1145/3371158.3371196>.
- [58] A. Petchiammal, B.K.S. Murugan, P. Arjunan, Paddy doctor: a visual image dataset for automated paddy disease classification and benchmarking, *IEEE Dataport* (2022), <https://doi.org/10.21227/hz4v-af08>.
- [59] S. Duhan, P. Gulia, N.S. Gill, P.K. Shukla, S.B. Khan, A. Almusharraf, N. Alkhaldi, Investigating attention mechanisms for plant disease identification in challenging environments, *Heliyon* 10 (9) (2024) e29802, <https://doi.org/10.1016/j.heliyon.2024.e29802>.
- [60] M. Tan, Q. Le, EfficientNetV2: Smaller Models and Faster Training, in: M. Meila, T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139 (2021) 10096–10106, (<https://proceedings.mlr.press/v139/tan21a.html>).
- [61] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, Image Is Worth 16x16 Words: Transform Image Recognit. Scale (2021), <https://doi.org/10.48550/arXiv.2010.11929>.
- [62] A. Singla, A. Nehra, K. Joshi, A. Kumar, N. Tuteja, R.K. Varshney, S.S. Gill, R. Gill, Exploration of machine learning approaches for automated crop disease detection, *Curr. Plant Biol.* 40 (2024) 100382, <https://doi.org/10.1016/j.cpb.2024.100382>.
- [63] A. Gamage, R. Gangahagedara, S. Subasinghe, J. Gamage, C. Guruge, S. Senaratne, T. Randika, C. Rathnayake, Z. Hameed, T. Madhujith, O. Merah, Advancing sustainability: the impact of emerging technologies in agriculture, *Curr. Plant Biol.* 40 (2024) 100420, <https://doi.org/10.1016/j.cpb.2024.100420>.
- [64] A. Upadhyay, N.S. Chandel, K.P. Singh, S.K. Chakraborty, B.M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, A. Elbeltagi, Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture, *Artif. Intell. Rev.* 58 (3) (2025) 92, <https://doi.org/10.1007/s10462-024-11100-x>.