Vol 8 No 1 (2026) *e*-ISSN: 2682-8383

**Journal of Engineering Technology and Applied Physics**

# Optimizer Performance in Deep Learning for Leaf Disease Classification

Obaid Shabbir[1], Sadaf Tanvir[1, *], Amjad Khan[1] and Nasir Gul[2]
[1]*Department of Computing, Abasyn University, Islamabad, Pakistan.*
[2]*Institute of Computer Science and IT, University of Science and Technology, Bannu, Pakistan.*
[*]*Corresponding author*: sadaf.tanvir@abasynisb.edu.pk, *ORCiD*: 0000-0002-3517-9405

https://doi.org/10.33093/jetap.2026.8.1.19
*Manuscript Received: 19 February 2025, Revised: 22 August 2025, Accepted: 20 December 2025, Published: 15 March 2026*

***Abstract*—Global food security is at risk due to the spread of plant diseases, particularly in regions where agriculture is a significant economic activity. Early and precise identification of plant diseases helps avoid crop loss and maintain agricultural output. Here, we investigate different deep learning optimizers for the classification of leaf diseases. We employ EfficientNet-B0, a cutting-edge model built on high accuracy and efficiency in image classification tasks intended for agricultural settings with limited resources. The PlantVillage and PlantDoc databases are used to determine the best optimizer. We evaluate the results of five popular optimizers on EfficientNet-B0: Adam, Nadam, Adagrad, RMSprop, and SGD. Empirical findings indicate that Adam produces training, validation, and testing outcomes that outperform other optimizers. Future real-time agricultural application implementations are anticipated to be fueled by this insight.**

***Keywords*—Deep learning, EfficientNet-B0, Leaf disease detection**

## I. INTRODUCTION

Deep learning has emerged as a promising approach in mitigating losses caused by plant diseases. They cause a substantial annual crop loss of around 10-40% worldwide [1]. In light of this, we provide a study that aims to highlight efficient and accurate deep learning models for classifying leaf diseases, thereby ensuring that crop losses are remediated in a timely manner. Although deep learning has demonstrated potential in the classification of diseases, the choice of optimizers in deep learning for disease detection has received little attention in these investigations. Most optimizers have been chosen based on best practices indicated in the literature. By utilizing the EfficientNet-B0 deep learning model and comparing its performance using various optimizers, our work closes this gap.

EfficientNet-B0 has stood out as one of the most effective models available for image categorization. EfficientNet-B0's compound scaling design balances network depth, width, and resolution. Compared to models like ResNet or DenseNet, this model trains with high accuracy using fewer parameters and at a lower computational cost. EfficientNet-B0 is an appropriate option because many agricultural applications need results in almost real time in settings with limited resources. However, the choice of optimizers in deep learning models has a significant impact on the accuracy, training stability, and convergence speed of models such as EfficientNet-B0.

Although deep learning for plant disease diagnosis shows promise, the effects of various optimizers on the performance of models like EfficientNet-B0 are mostly unknown. While poor choices may affect the outcome, the correct optimizer selection can enhance model performance. Here, we bridge this gap by methodically assessing the effectiveness of a number of well-known optimizers, identifying the top and bottom-performing optimizers for the identification of leaf disease. With the goal of facilitating the efficient diagnosis of leaf diseases, this work offers insights for further investigation and real-world applications of optimal optimizers. The sequential stages and technique used in this investigation are depicted in the graphical abstract in Fig. 1. The procedure from data collection and preprocessing to model training and evaluation is outlined in the diagrammatic overview that follows. The related works and motivation are presented in Section II and Section III. Methodology is described in Section IV. The results, their analysis and discussion are covered in Section V. The paper's conclusions are presented in Section VI.

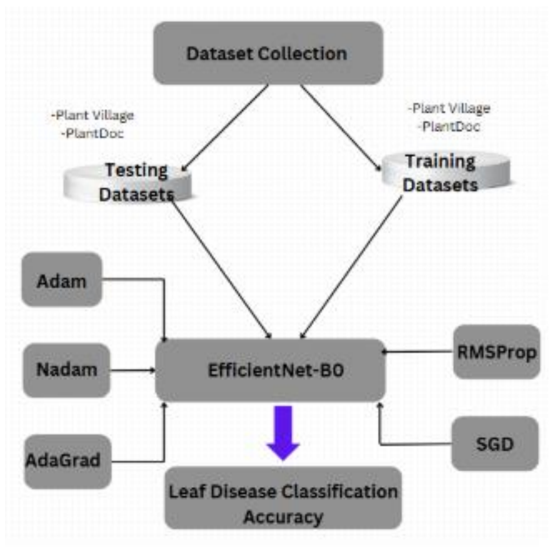

Fig. 1. Graphical abstract.

## II. RELATED WORK

Deep learning models for visual plant disease diagnosis are currently an active area of research. Classical classifiers and manually created features were employed in earlier models. Convolutional neural networks (CNNs) have been employed recently by researchers for feature extraction and categorization. For example, on a fresh dataset of more than 50,000 photos, Konduru *et al.* [2] employed CNN architectures like AlexNet and MobileNet to identify and categorize plant illnesses. Building on these developments, Yang *et al.* [3] achieved state-of-the-art results on the AI Challenger dataset by proposing a novel CNN and transformer combination based on a multi-label technique to simultaneously detect plant species, leaf diseases, and their severity. Similar research employing CNN-based plant disease categorization techniques was also carried out by Saleem *et al.* [4].

A new standard for model correctness and efficiency was set with the release of EfficientNet, more specifically EfficientNet-B0. In image classification tasks, balancing network depth, width, and resolution using compound scaling showed notable performance improvements while preserving computing economy. As a result, EfficientNet-B0 has been used in numerous studies to detect plant diseases. For example, EfficientNet's use in agricultural picture classification is covered by Farman *et al.* [5], while EfficientNet-B0 is also used for plant disease detection by Fathimathul *et al.* [6] and Alqahtani *et al.* [7]. Our work evaluates the effects of several optimizers on EfficientNet's performance in plant disease classification.

Neural network training time, convergence, and accuracy are influenced by optimizers such as SGD, Adam, Nadam, RMSprop, and Adagrad in deep learning. Studies show that in EfficientNet-B0, optimizers like Nadam, RMSProp, and Adagrad are less commonly employed than SGD and Adam.

In deep learning, optimizers like SGD, Adam, Nadam, RMSprop, and Adagrad influence training time, convergence, and accuracy of neural networks. While SGD and Adam are very widely used, optimizers such as Nadam, RMSProp, and Adagrad are less frequently used. For example, RMSprop is less commonly applied because it can handle non-stationary data or data with high variability, which is not always the case [8]. Similarly, Adagrad is designed for sparse data handling and is less common in scenarios where models are fine-tuned on large pre-trained datasets such as ImageNet [9]. Alternatively, Nadam is less used since it is sensitive to hyperparameter tuning, and performance gains over Adam are usually marginal for image classification tasks.

Nonetheless, optimizers in deep learning have not been an extensive subject of research. For instance, we see a few studies like Wilson *et al.* [10] which examines the convergence and performance of several optimizers in deep learning models, while Sun [11] provides an overview of optimizers and their theoretical background. Another study about optimizers and model performance is presented by Dogo *et al.* [12]. By contrasting the effects of well-known optimizers on EfficientNet-B0 for visual leaf disease identification, we expand on these investigations using real field image datasets like the PlantDoc dataset [13].

The choice of EfficientNet is justified by the fact that it has outperformed other contemporary architectures in terms of computing. In order to detect plant diseases, researchers compared EfficientNet to other CNN architectures like ResNet, DenseNet, and MobileNet. For instance, Kunduracıoğlu *et al.* [14] employs EfficientNet-B0 to identify sugarcane leaf disease, whereas Duong *et al.* [15] demonstrates that EfficientNet outperforms ResNet for agricultural jobs. Different architectures are the subject of these comparisons. In this study, we examine the effects of various optimizers on a single architecture, EfficientNet-B0, in the less-studied field of leaf disease detection.

Plant disease detection is also being studied using federated learning and transfer learning. For example, Shafik *et al.* [16] and Khan *et al.* [17] use transfer learning, and Choubey *et al.* [18] use federated transfer learning to study plant disease detection. Optimizer performance is enhanced by enabling faster convergence and avoiding overfitting by fine-tuning pre-trained EfficientNet models on ImageNet weights. In particular, momentum-based fine-tuners like Adam and SGD preserve important generic characteristics while adjusting to the task of detecting plant diseases.

Our study advances the state-of-the-art techniques for agricultural disease diagnosis by enhancing model accuracy and efficiency through testing the above stated optimizers.

## III. MOTIVATION AND PROBLEM STATEMENT

The increasing reliance on deep learning for plant disease detection highlights the need for efficient and accurate models. While EfficientNet-B0 has proven effective, the impact of various optimizers on its performance remains underexplored. Understanding how optimizers like SGD, Adam, and RMSprop affect accuracy and training efficiency in plant disease classification for real field image datasets could lead to better model performance, advancing precision agriculture and supporting sustainable farming practices. Current work on optimizers in deep learning present significant knowledge gaps in understanding their performance across different architectures. Although these studies show that adaptive optimizers like ADAM and NADAM excel in performance [12] and [13], they provide limited insights explaining why certain optimizers consistently outperform others in specific contexts. Also, how optimizer dynamics evolve throughout the training process remain largely unexplored. Recent research often employs default hyper parameter settings and its focus is on image classification tasks, overlooking how diverse dataset characteristics might influence optimizer effectiveness. This study aims to fill this gap by evaluating how various optimizers affect architectural choices, dataset properties, and learning dynamics across leaf disease detection. This study can provide guidelines to practitioners with principled guidelines for optimizer selection rather than relying on trial-and-error approaches.

## IV. METHODOLOGY

This section discusses the methodology adopted to perform this study.

### *A. Dataset Description and Analysis*

This work employs two publicly available datasets: Plant Village & Plant Doc. Image labels of healthy and diseased crops from the Plant Village dataset (Fig. 2) include potatoes, maize, and tomato. The more diverse and challenging PlantDoc dataset (Fig. 3) provides annotated images of plant diseases taken in real-life conditions. The dataset, called "Plant Village," came from [19]. It was selected because it was relevant to our research question - the classification of corn, potato, and tomato healthy and diseased leaves. Tables I - IV show dataset details for potato, corn and tomato leaf diseases.

Table I. Potato dataset description.

| *Potato Healthy* | *Early Blight* | *Late Blight* | *Total* |
|---|---|---|---|
| 500 | 500 | 500 | 1500 |

Table II. Corn dataset description.

| *Corn Healthy* | *Cercospa Leaf spot* | *Northern Leaf blight* | *Common Rust* | *Total* |
|---|---|---|---|---|
| 4648 | 2052 | 4940 | 4768 | 16408 |

Table III. Tomato dataset description.

| *Healthy* | *Bacterial Spot* | *Early Blight* | *Late Blight* | *Leaf Mold* | *Septoria Leaf Spot* | *Spider Mite* | *Target Spot* | *Mosaic Virus* | *Leaf curl Virus* | *Total* |
|---|---|---|---|---|---|---|---|---|---|---|
| 1591 | 2127 | 1000 | 1908 | 952 | 1771 | 1676 | 1404 | 373 | 5357 | 18159 |

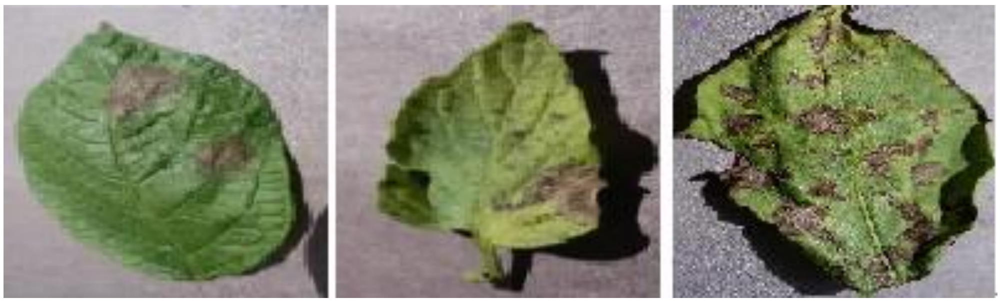
Fig. 2. (a) Potato Late Blight, (b) Tomato Early Blight, (c) Potato Early Blight-Samples from PlantVillage dataset (www.kaggle.com [18]).

Table IV. Plant Village dataset description.

| | |
|---|---|
| Total Images | 36067 |
| Training | 23055 |
| Validation | 5341 |
| Testing | 7671 |

"PlantVillage" dataset contains many images of plant leaves with different diseases suitable for our classification task. This dataset has three subsets: 80%, 10%, and 10% respectively of the training set, validation set, and test set are distributed. Details are shown in Table V. The images were standardized and prepared for analysis before feeding them into the deep learning model. These pre-processing steps included: Resizing, normalization, and data augmentation, adjustment of brightness range, rotation, horizontal and vertical flips. Utilizing pre-trained features, transfer learning was applied to the EfficientNet-B0 architecture. This knowledge gained from the ImageNet dataset is used to train the model to extract relevant features from plant leaf images.

The PlantDoc Dataset consists of 2,598 images that are classified into 13 different plant species and 27 distinct plant diseases. This dataset is split into 2,165 training images and 433 testing images. The images depict plants in various real-world environments, including diverse lighting and backgrounds, making it suitable for building robust plant disease detection models. The dataset aims at supporting model development for both healthy plants and those with various diseases. It is especially applicable to agriculture where disease classification can help with disease prevention and crop protection strategies. Tables VI - VIII show the details of the PlantDoc dataset for potato, corn and tomato.

Table V. PlantDoc dataset description.

| Total Images | 2598 |
|---|---|
| Training | 2165 |
| Validation | – |
| Testing | 433 |

Table VI. Potato dataset description-PlantDoc.

| *Potato Healthy* | *Potato Early Blight* | *Potato Late Blight* | *Total* |
|---|---|---|---|
| 400 | 100 | 100 | 600 |

Table VII. Corn dataset description-PlantDoc.

| *Healthy* | *Cercospora Leaf spot* | *Northern Leaf blight* | *Common Rust* | *Total* |
|---|---|---|---|---|
| 104 | 92 | 93 | 96 | 385 |

Table VIII. Tomato dataset description- PlantDoc.

| *Healthy* | *Bacterial Spot* | *Early Blight* | *Late Blight* | *Leaf Mold* | *Septoria Leaf Spot* | *Spider Mite* | *Target Spot* | *Mosaic Virus* | *Yellow Leaf Curl Virus* | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| - | 141 | 135 | 135 | 66 | 67 | 77 | 70 | 62 | 278 | 1031 |

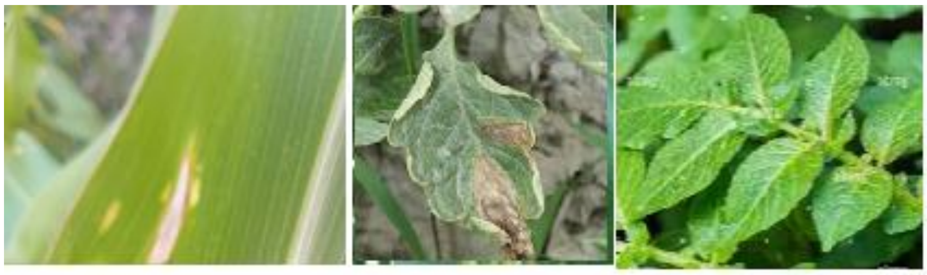

Fig. 3. (a) Corn Grey Leaf Spot, (b) Tomato Late Blight, (c). Potato Healthy-Plant Doc dataset samples (www.kaggle.com [18]).

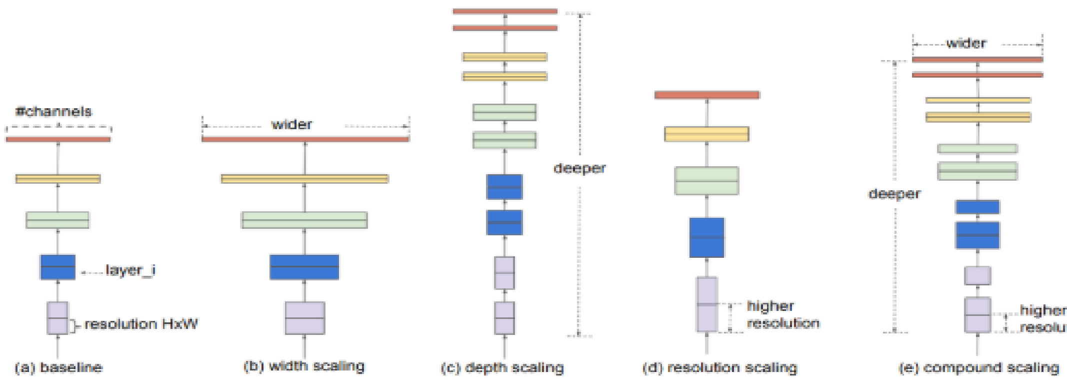

Fig. 4. Architecture of EfficientNet-B0.

### *B. Pre-processing and Model Architecture*

Images for both datasets were resized to a uniform dimension of 224×224 pixels for compatibility with the model input size. Images were augmented with vertical flipping and rotation to improve the variability and robustness of the dataset. Pixel values were normalized to 0-1.

For image classification tasks, we used EfficientNet-B0 architecture (Fig. 4) because it is very fast and efficient on the computation side. Pre-trained ImageNet weights were used as a base, and the last layer was replaced by a fully connected layer representing the number of plant species in the datasets.

The fundamental architecture of EfficientNet-B0 rests on compound scaling, balancing three factors - depth, width, and resolution of the neural network. Key components of EfficientNet-B0 are listed below:

1. Input Layer: The input to EfficientNet-B0 is an image resolution of 224 × 224 × 3 (for RGB images).

2. The stem layer is the first convolutional layer applying 3×3 convolution with 32 filters, followed by a Batch Normalization Layer and a Swish activation function. This layer reduces the input spatial dimensions.

3. EfficientNet-B0 uses a type of depthwise separable convolution called Mobile Inverted Bottleneck Convolution Blocks. These represent basic network elements, which are staged in different stages of implementation: MBConv1 (Expansion factor 1, SE ratio 0.25) - Initial layers. For later layers, use MBConv6 (Expansion factor 6, SE ratio 0.25). A typical MBConv block contains:

(i) Expansion phase: Increase the number of channels (features).

(ii) Depthwise convolution: A lightweight convolution operation on each input channel.

(iii) Squeeze-and-Excitation (SE) block: This mechanism recalibrates the feature channels so the network can focus on important features.

(iv) Projecting phases: Project the features back to fewer channels.

(v) Residual connection added if the input and output dimensions are equal.

4. Stages in EfficientNet-B0: The network has several stages that increase the depth, width (channels), and resolution:

(i) Stage 1: 1 MBConv1 block with 16 filters.

(ii) Stage 2: 2 MBConv6 blocks with 24 filters (expansion factor 6).

(iii) Stage 3: 2 MBConv6 blocks with 20 filters.

(iv) Stage 4: 3 MBConv6 blocks containing 80 filters.

(v) Stage 5: 3 MBConv6 blocks with 112 filters.

(vi) Stage 6: 4 MBConv6 blocks with 192 filters.

(vii) Stage 7: 1 MBConv6 with 320 filters.

5. After the MBConv blocks are done, a 1 × 1 convolutional layer with 1280 filters is applied. Then a global average pooling layer is applied to reduce the spatial dimensions. A fully connected (dense) layer of 1000 neurons is applied for classification in ImageNet. For final class probabilities, the output is passed through a softmax activation.

6. Swish activation function: EfficientNet uses Swish Activation Function instead of ReLU.

7. EfficientNet-B0 uses compound scaling, where the network depth, width, and resolution are scaled uniformly. In B0 the depth multiplier is 1.0, width multiplier is 1.0, and resolution is 224 × 224.

### *C. Optimizers*

We tested five optimizers: Adam, Nadam, Adagrad, RMSprop, and SGD. Given below in Table IX, we compare these five optimizers and their relevance in deep learning for leaf disease detection. This table has been generated by the facts widely published in the literature.

Although less explored, optimizers RMSProp, Nadam, and Adagrad are listed below, they may be applied to the following application scenarios: A leaf disease dataset might have significant variability and nonstationary characteristics because of environmental conditions (lighting/backgrounds), leaf orientation/position (different angles/zoom levels), and disease progression/severity (early vs. late stages).

Also, variation in leaf shape among species, health statuses, and conditions can impair detection. When data are collected over time and plant diseases change due to seasonal variations, weather conditions, or growth cycles occur. All these variations impair the generalization of the model and make the choice of optimizer critical.

Nadam could be an appropriate choice for optimizing EfficientNet-B0 in some situations, such as when working with non-stationary data where its Nesterov momentum helps the model cope with rapid changes or when faster convergence is needed. Also, it can be useful for deep fine-tuning tasks where exact changes to pre-trained weights are required or when working with sparse datasets where its adaptive learning rate and momentum can improve performance. Also, Nadam's ability to overcome local minima might prove useful in more complicated optimization landscapes, especially for specialized tasks with small differences in data. But Nadam often requires careful hyperparameter tuning, and is thus a useful option when fine-tuning settings is possible. Similarly, Adagrad may be justified for EfficientNet-B0 when dealing with sparse data, tasks requiring automatic learning rate adaptation, or when long-term training stability is required. This gives a robust solution for data imbalance or rare feature problems.

The same architecture and hyper parameters were used for testing each optimizer:

- Learning rate: Set the learning rate to 0.01 for all optimizers.
- Batch size: 32.
- Number of epochs: 10 for each experiment.

Table IX. Comparison of optimizers for leaf disease detection.

| Optimizer | Description | Strengths | Limitations | Relevance to Leaf Disease Detection |
|---|---|---|---|---|
| **Adam** | Combines advantages of Adagrad and RMSprop; adapts learning rate for each parameter individually. | Fast convergence, works well for a range of problems, reduces need for manual learning rate tuning. | Prone to overfitting, may require more computational resources. | Highly relevant due to strong generalization and high accuracy in all phases (training, validation, testing). |
| **Nadam** | Variant of Adam incorporates Nesterov momentum for potentially faster convergence. | Fast convergence, incorporates momentum and adaptive learning rate. | Can overfit, sensitive to hyperparameters. | Moderately relevant for similar performance to Adam but faster convergence. |
| **Adagrad** | Adjusts learning rate for each parameter based on past gradients; good for sparse data. | Good for sparse data, adapts learning rate automatically. | Learning rate can decay too quickly, poor performance on large datasets. | Less relevant as it underperforms on large datasets like Plant Village. |
| **SGD** | Updates parameters based on gradient of loss function with momentum for faster convergence. | Simple, effective for large-scale problems with momentum, high efficiency in large networks. | Slow convergence without momentum, requires careful learning rate tuning. | Highly relevant due to simplicity and effectiveness in large datasets with high accuracy. |
| **RMSprop** | Adapts learning rate by dividing by root mean square of recent gradient magnitudes; good for non-stationary objectives. | Good for non-stationary objectives, helps prevent vanishing learning rates. | May overfit, requires careful tuning of decay rate. | Moderately relevant for tasks with non-stationary patterns, though can overfit. |

### *D. Training and Validation*

The datasets were split into 70:15:15 training, validation, and test sets. The model parameters were optimized using the training set and the hyperparameters were fine-tuned using the validation set. For each optimization capability, generalization capability was evaluated, accuracy was recorded during training and validation phases.

### *E. Performance Metrics*

Performance of each optimizer was evaluated based on accuracy. Training and validation accuracy for all optimizers were recorded at every epoch. The final testing accuracy was also computed on the unseen test set once the model was trained and validated.

## V. RESULTS AND DISCUSSION

We assess the effectiveness of five optimizers (Adam, Nadam, Adagrad, SGD, and RMSprop) on EfficientNet-B0. Adam exhibits almost flawless accuracy for potato during the training, validation, and testing stages in Fig. 5, and SGD also does well. Compared to Adam and SGD, Nadam and Adagrad do not perform well while achieving accuracy. RMSprop performs poorly across the board in terms of accuracy. Adam and Nadam exhibit overfitting in Fig. 6 with high training accuracy but low validation and testing accuracy. RMSprop also exhibits overfitting symptoms as it performs badly on test data while having excellent training and validation scores. Adam and SGD achieve accuracies of between 0.9 and 0.95 for maize and tomato crops (Fig. 6 and Fig. 7).

RMSprop attains the best training accuracy which is depicted in Figs. 8, 9 and 10 using the Plant Doc dataset, followed by Nadam and Adam. Validation accuracy suggests overfitting, which remains constant at 0.5 across optimizers. Test generalization is stronger for RMSprop and SGD at 0.531 than for the others, which are at 0.468. RMSprop's versatility allows it to handle complex datasets with noise while maintaining reasonable generalization and training accuracy. Adam has the best training accuracy (0.5938) in Fig. 9, but it appears to be overfitting. In Fig. 10, with balanced training and test performance, Adam is the best overall, whereas RMSprop and Nadam overfit with high training but low test accuracy. RMSprop, Nadam, and Adagrad need to be adjusted to prevent overfitting, whereas Adam is the recommended optimizer overall, with SGD as a possible substitute. It is to note that in Fig. 10, the significant drop in validation and testing accuracy points to either the optimizer-model compatibility in the form of vanishing gradient, inappropriate learning rates, or numerical instability where the model fails to generalize for unseen data. Although some optimizers perform better in this context because they effectively balance learning rates and adapt to the dataset's structure, allowing the model to converge toward meaningful patterns rather than overfitting. We need to improve hyperparameter tuning and apply regularization strategies to enhance generalization.

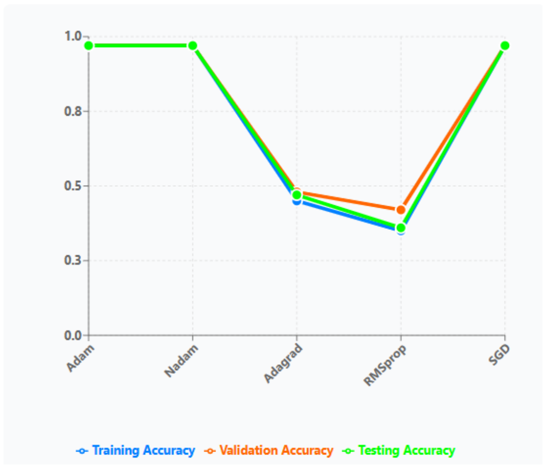

Fig. 5. Accuracy for potato using Plant Village dataset.

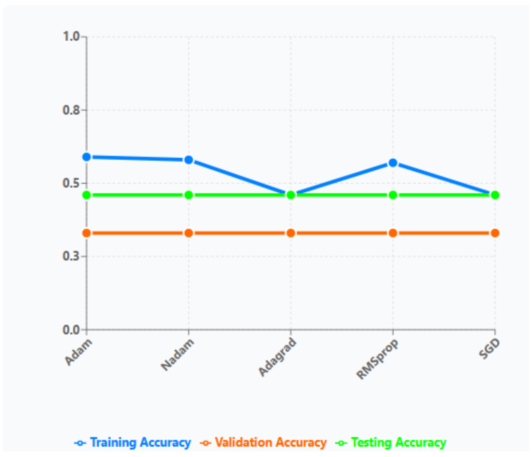

Fig. 8. Accuracy for potato using Plant Doc dataset.

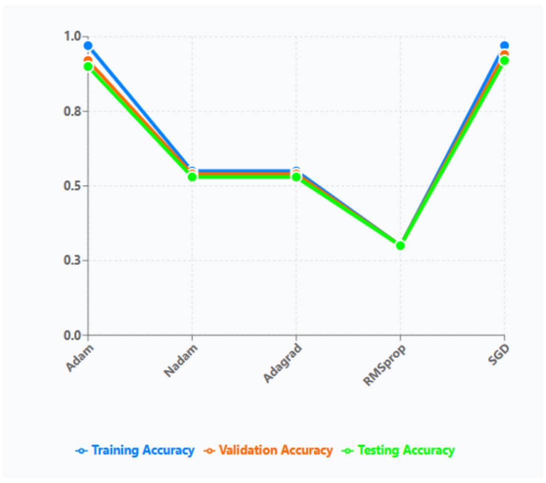

Fig. 6. Accuracy for maize using Plant Village dataset.

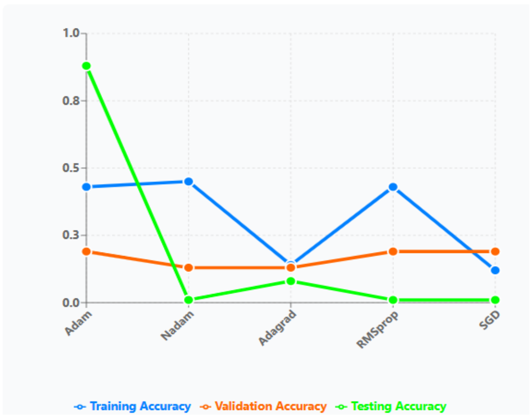

Fig. 9. Accuracy for maize using Plant Doc dataset.

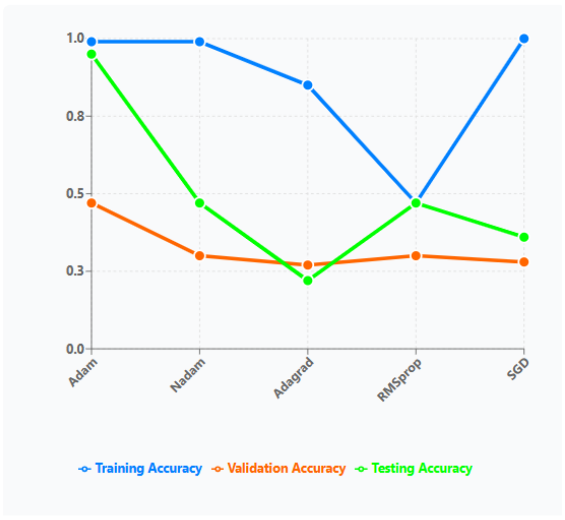

Fig. 7. Accuracy for tomato using Plant Village dataset.

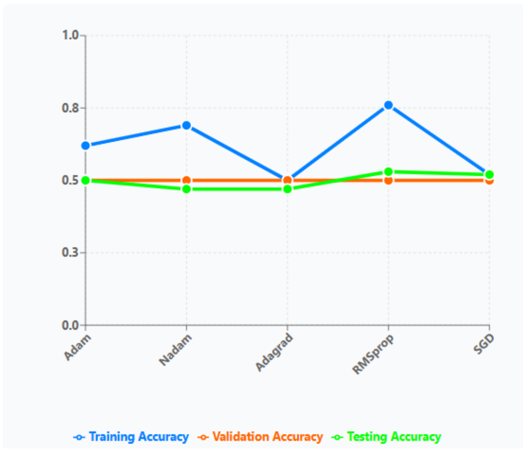

Fig. 10. Accuracy for tomato using Plant Doc dataset.

## VI. CONCLUSION

Using prominent optimizers, we investigated EfficientNet-B0's performance to classify diseases of plant leaves mainly in potato, maize, and tomato. For the three crops, we have observed empirically that Adam is the best optimizer in the Plant Village and Plant Doc dataset because it can adapt the learning rate for each parameter to achieve faster and more efficient convergence. We do have limitations though. A somewhat limited dataset that we have used might have yielded these results. More real-life datasets need to be explored. Furthermore, study can be tuned further using additional hyper parameters to observe further performance gains. Also, emerging technologies like drones and the electromagnetic spectrum may be studied to help farmers monitor plants better and collect continuous data. Having input from experts in agriculture and refining our model are important steps.

## ACKNOWLEDGEMENT

The authors would like to thank anonymous reviewers for their helpful comments in improving the quality of this manuscript.

## FUNDING STATEMENT

The authors did not receive any funding for the research materials of this article.

## AUTHOR CONTRIBUTIONS

Obaid Shabbir: Software Development and implementation, Running Experiments,
Sadaf Tanvir: Conceptualization, Supervision, Technical Writing, Experimental Design, Methodology, Investigation, Validation, Writing, Review, Editing and Manuscript Refinement,
Amjad Khan: Project Administration, Technical Guidance, Technical Review & Editing,
Nasir Gul: Methodology Design Improvement, Investigation, Validation, Review and Editing.

## CONFLICT OF INTERESTS

No conflict of interests was disclosed.

## ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

## REFERENCES

[1] Y. Gai and H. Wang, "Plant Disease: A Growing Threat to Global Food Security," *Agronomy*, vol. 14, no. 8, pp. 1615, 2024.

[2] S. Konduru, M. Amiruzzaman, V. D. Avina and M. R. Islam, "Plant Disease Detection and Classification Using Deep Learning Models," *J. Comput. Sci. Colleg.*, vol. 39, no. 3, pp. 66-75, 2023.

[3] B. Yang, M. Li, F. Li and H. Wang, "A Novel Plant Type, Leaf Disease, and Severity Identification Framework Using CNN and Transformer with Multi-label Method," *Sci. Rep.*, vol. 14, no. 1, pp. 116644, 2024.

[4] M. H. Saleem, J. Potgieter and K. M. Arif, "Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers," *Plants*, vol. 9, no. 10, pp. 1319, 2020.

[5] H. Farman, J. Ahmad, B. Jan, Y. Shahzad, M. Abdullah and S. Ali, "EfficientNet-based Robust Recognition of Peach Plant Diseases in Field Images," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 2073–2089, 2022.

[6] P. P. Fathimathul Rajeena, S. U. Aswathy, M. A. Moustafa and M. A. S. Ali, "Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach," *Electron.*, vol. 12, no. 8, pp.1938, 2023.

[7] M. M. Alqahtani, A. K. Dutta, S. Almotairi, M. Ilayaraja, A. A. Albraikan, F. N. Al-Wesabi and M. Al Duhayyim, "Sailfish Optimizer with EfficientNet Model for Apple Leaf Disease Detection," *Comput., Mater. & Contin.*, vol. 75, no. 1, pp. 217-232, 2023.

[8] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv,* vol. abs/1609.04747, 2016.

[9] J. Heaton, "Ian Goodfellow, Yoshua Bengio and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618," *Genet. Program. Evolvable Mach.*, vol. 19, no. 1, pp. 305 – 307, 2018.

[10] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," *Adv. Neur. Informat. Process. Syst.*, vol. 30, 2017.

[11] R. Y. Sun, "Optimization for Deep Learning: An Overview," *J. the Operat. Res. Soc. of China*, vol. 8, no. 2, pp. 249-294, 2020.

[12] E. M. Dogo, O. J. Afolabi and B. Twala, "On the Relative Impact of Optimizers on Convolutional Neural Networks with Varying Depth and Width for Image Classification," *Appl. Sci.*, vol. 12, no. 23, pp. 11976, 2022.

[13] P. Singh, P. Singh, U. Farooq, S. S. Khurana, J. K. Verma and M. Kumar, "CottonLeafNet: Cotton Plant Leaf Disease Detection Using Deep Neural Networks," *Multimed. Tools Appl.*, vol. 82, no. 24, pp. 37151–37176, 2023.

[14] İ. Kunduracıoğlu and İ. Paçal, "Deep Learning-based Disease Detection in Sugarcane Leaves: Evaluating EfficientNet Models," *J. Oper. Intell.*, vol. 2, no. 1, pp. 321–335, 2024.

[15] L. T. Duong, P. T. Nguyen, C. Di Sipio and D. Di Ruscio, "Automated Fruit Recognition using EfficientNet and MixNet," *Comput. Electron. Agric.*, vol. 171, pp. 105326, 2020.

[16] W. Shafik, A. Tufail, C. De Silva Liyanage and R. A. A. H. M. Apong, "Using Transfer Learning-based Plant Disease Classification and Detection for Sustainable Agriculture," *BMC Plant Bio*., vol. 24, no.1, pp. 136, 2024.

[17] I. Khan, S. S. Sohail, D. Ø. Madsen and B. K. Khare, "Deep Transfer Learning for Fine-Grained Maize Leaf Disease Classification," *J. Agric. Food Res.*, vol. 16, pp. 101148, 2024.

[18] S. Choubey and Divya, "Lightweight Federated Transfer Learning for Plant Leaf Disease Detection and Classification Across Multiclient Cross-Silo Datasets," in *BIO Web of Conf.*, vol. 82, pp. 05018, 2024.

[19] S. P. Mohanty, D. P. Hughes and M. Salathé, "Using Deep Learning for Image-based Plant Disease Detection," *Front. Plant Sci.*, vol. 7, pp. 1419, 2016.