https://dx.doi.org/10.4314/ijs.v26i3.1

HARNESSING DEEP LEARNING ALGORITHMS FOR EARLY PLANT DISEASE DETECTION: A COMPARATIVE STUDY AND EVALUATION BETWEEN SSD (MOBILENET_V2 AND MOBILENET_V3) AND CNN MODEL

Nwaneto, C. B. and Yinka-Banjo, C.*
Department of Computer Sciences, University of Lagos, Akoka, Lagos, Nigeria.
*Corresponding Author's Email: cyinkabanjo@unilag.edu.ng
(Received: 31st July, 2024; Accepted: 10th December, 2024)

ABSTRACT

Recognizing the important need for efficient plant disease detection in agriculture, this research evaluates and compares the performance of three distinct deep learning models: Mobilenet_V2, Mobilenet_V3, and a custom-built CNN model. As traditional methods fall short in addressing the evolving challenges of crop health management, the study aims to discover the most effective model for accurate disease identification. Leveraging a dataset encompassing 20,639 images across 15 directories representing various plant diseases, the models undergo rigorous training and evaluation. Results reveal the CNN_model as the superior performer with a remarkable test accuracy of 94.48%, outshining Mobilenet_V2 and Mobilenet_V3. The comparative analysis sheds light on the strengths and weaknesses of each model, providing valuable insights for the agricultural community. This research not only advances the understanding of deep learning applications in precision agriculture but also lays the foundation for future innovations in sustainable crop management.

Keywords: Plant Disease Detection; Deep Learning Models; Convolutional Neural Networks (CNN); MobileNet_V2 and MobileNet_V3; Precision Agriculture; Machine Learning in Agriculture.

INTRODUCTION

The agriculture sector plays a pivotal role in sustaining global food security, making the timely detection and management of plant diseases crucial for ensuring optimal crop yield (Shiferaw *et al.*, 2013). As technology continues to advance, machine-learning models have emerged as powerful tools for automating the process of plant disease detection, offering a promising solution to the challenges faced by traditional methods (Singh *et al.*, 2016). The timely and accurate detection of these diseases is crucial for effective management and mitigation strategies. This research delves into the comparison of three deep learning models, specifically Mobilenet_V2 and Mobilenet_V3 which are both types of Single Shot Detectors, along with a custom-built Convolutional Neural Network (CNN) model, to determine the most efficient of them which can enhance the early detection of plant diseases.

Despite advancements in technology, plant diseases continue to impact crop health, leading to substantial economic losses and food shortages. Traditional methods of disease detection are often time-consuming and prone to human error (Arsenovic *et al.*, 2019). The need for efficient and automated solutions has prompted the exploration of deep learning algorithms. This research addresses the existing gap in understanding the comparative performance of popular pre-trained models and custom-built CNN models for plant disease detection, focusing on performing experiments using a diverse dataset to evaluate the performance of Mobilenet_V2, Mobilenet_V3, and a custom-built CNN model for plant disease detection, and analysing the results to determine the most effective model for early detection.

Plant Disease Detection

Farmers face a primary challenge in dealing with crop diseases, making the classification and analysis of these illnesses pivotal for optimizing food yield in agriculture (Chen *et al.*, 2021). The study of detecting and recognizing plant disease is vital, especially as it can potentially monitor extensive crop fields and promptly identify disease symptoms on plant leaves (Martinelli *et al.*, 2015). Therefore, the quest for a quick, efficient, cost-effective, and effective approach to determining instances of crop diseases is of utmost importance (Chen *et al.*, 2021).

Artificial intelligence (AI) significantly contributes to the agricultural sector, enhancing a nation's

gross domestic product (GDP). Applying edge intelligence to agriculture, particularly through the use of deep learning models like the YOLOv3 neural network on embedded systems such as the NVIDIA Jetson TX2, presents a novel approach. This involves implementing the system on a drone to capture plant images, identify pest positions, and apply pesticides as needed (Al-Hiary *et al.*, 2011).

Hyperspectral and multispectral knowledge acquisition techniques have proven valuable in improving agricultural production by providing crucial data on elements affecting crop condition and growth. Widely employed in various agricultural applications, including sustainable agriculture, these technologies offer essential insights for farmers and agricultural management (Ang, 2021).

Deep Learning in Agriculture

Predicting diseases at an early stage is crucial to preventing massive crop loss and ensuring higher crop production. Advances in high computing speed and power, along with improved access to massive datasets, contribute to the enhanced efficiency of disease detection systems (Wei *et al.*, 2020). A hybrid combination of classifier techniques demonstrated a recognition rate of 91.11%, surpassing the performance of serial, parallel, and deep learning approaches (Massi *et al.*, 2020).

Additionally, the application of the CovN*et al*gorithm for identifying weeds in crops and the use of hybrid deep-learning models demonstrate promising results in terms of accuracy and parameter reduction (Zhou *et al.*, 2021). Deep transfer learning, utilizing pre-trained datasets like Inception and ImageNet modules, proves effective in identifying plant diseases, with support vector machine (SVM) and multi-layer perceptron contributing to higher accuracy. The deployment of Deep Convolutional Neural Networks for recognizing corn dietary sickness and the use of hardware enhancements, such as Raspberry Pi3 with an Intel Movidius Neural Compute Stick, has resulted in superior metric accuracy performance (Sun *et al.*, 2020).

As AI and DL technology continue to grow, computer vision (CV) has made significant strides. CV-based approaches, particularly those employing principle component analysis (PCA) and backpropagation methods, have proven useful in diagnosing grape leaf diseases with high research accuracy (Xie *et al.*, 2020). Validation accuracy using VGGNet on real-world datasets reached 91.83% (Xie *et al.*, 2020).

The models are applied to object detection and semantic segmentation tasks, with a new efficient segmentation decoder called Lite Reduced Atrous Spatial Pyramid Pooling (LR-ASPP) proposed for semantic segmentation. MobileNetV3-Large achieves 3.2% higher accuracy on ImageNet classification with a 20% reduction in latency compared to MobileNetV2 (Sandler *et al.*, 2018). MobileNetV3-Small is 6.6% more accurate than a comparable MobileNetV2 model with similar latency. MobileNetV3-Large also demonstrates over 25% faster detection on COCO detection compared to MobileNetV2, and MobileNetV3-Large LR-ASPP is 34% faster than MobileNetV2 R-ASPP with similar accuracy for Cityscapes segmentation (Howard *et al.*, 2019). Table 1 shows a comparison between Mobilenet_V2 and Mobilenet_V3 architectures.

Howard *et al.*, (2019) emphasize the importance of efficient neural networks for on-device experiences, personal privacy, and battery life preservation. The development of MobileNetV3 Large and Small models is aimed at delivering high-accuracy, efficient neural network models for on-device computer vision. The models showcase a trade-off between accuracy and latency, pushing the state of the art forward and highlighting the effectiveness of combining automated search with novel architecture advancements.

Table 1: Comparison of Mobilenet_V2 and Mobilenet_V3

FEATURE	Mobilenet_V2	Mobilenet_V3
ACTIVATION FUNCTION	ReLU	Hardswish
RESIDUAL CONNECTION	Standard residual	Inverted residual
ATTENTION MECHANISM	None	SE modules
TOP-1 ACCURACY (IMAGE NET)	72.00%	75.20%
PARAMS	3.4M	5.4M
MACS	300M	560M

MobileNet_V2

MobileNet_V2 uses a combination of depth-wise separable convolutions and inverted residuals to achieve both efficiency and accuracy (Chirasani *et al.*, 2024). Depth-wise separable convolutions factorize a standard convolution into two separate operations: depth-wise convolution and pointwise convolution. This factorization reduces the number of parameters and computations required for each convolution, making the network more efficient. Inverted residuals are a type of residual connection that inverts the order of the operations in a standard residual connection. This inversion allows for the use of linear bottlenecks, which further reduce the number of parameters and computations required for each residual connection (Howard *et al.*, 2017).

Mobilenet_V2 Architecture

The fundamental building block of this model consists of a bottleneck depth-separable convolution with residuals, offering a refined structure as seen in Figure 1 and outlined in ***Table 2***. This architecture is designed to transform input tensors from k to k' channels, with a specific stride (s) and an expansion factor (t).

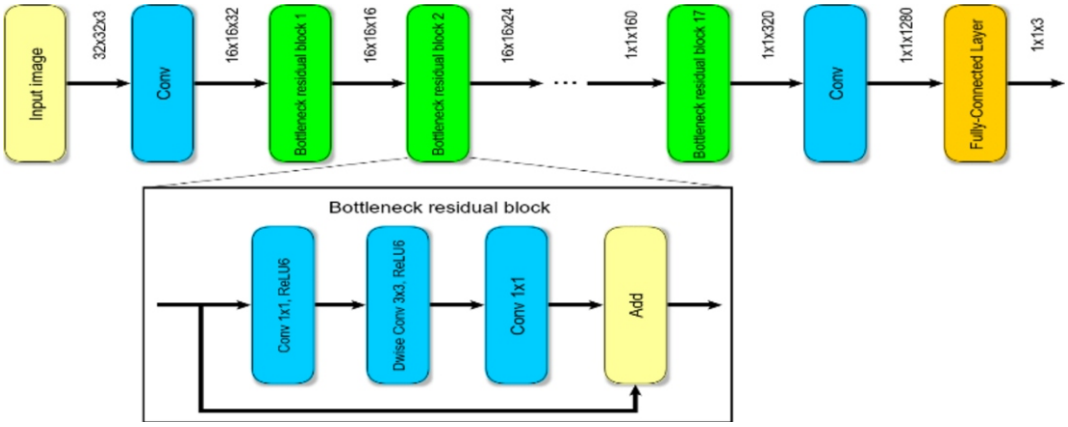

Figure 1: Architectural diagram of MobileNetV2.

Table 2: Structure of the bottleneck residual block, emphasizing its role in channel transformation.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

The overall MobileNetV2 architecture encompasses an initial fully convolutional layer featuring 32 filters. This is succeeded by 19 residual bottleneck layers, as illustrated in Table 2. ReLU6 is employed as the non-linearity due to its resilience in low-precision computations (Howard

et al., 2017). Throughout the network, a standard 3x3 kernel size is consistently utilized, and training incorporates dropout and batch normalization.

Except for the initial layer, a constant expansion rate is maintained across the network. Through experimentation, expansion rates ranging from 5 to 10 yield comparable performance, with smaller networks benefiting from slightly reduced expansion rates and larger networks showcasing improved performance with larger expansion rates.
In the primary experiments, Howard *et al* employ an expansion factor of 6, applied to the size of the input tensor. For instance, for a bottleneck layer transforming a 64-channel input tensor into one with 128 channels, the intermediate expansion layer comprises $64 \times 6 = 384$ channels.

Howard *et al* (2017), enable customization of the architecture for different accuracy/performance trade-offs. Tunable parameters include input image resolution and width multiplier.

The cost of computation of the primary network, with a width multiplier of 1 and a resolution of 224×224, is 300 million multiply-adds, utilizing 3.4 million parameters. Howard *et al*, explored various performance trade-offs by adjusting input resolutions from 96 to 224 and width multipliers from 0.35 to 1.4. Computational costs range from 7 million to 585 million multiply-adds, while model sizes vary between 1.7 million and 6.9 million parameters.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 3. The architecture of MobileNetV2, illustrating a sequence of identical layers repeated n times, with each layer having the same number of output channels c (Howard *et al.*, 2017).

Size	MobileNetV1	MobileNetV2	ShuffleNet (2x,g=3)
112x112	1/O(1)	1/O(1)	1/O(1)
56x56	128/800	32/200	48/300
28x28	256/400	64/100	400/600K
14x14	512/200	160/62	800/310
7x7	1024/199	320/32	1600/156
1x1	1024/2	1280/2	1600/3
max	800K	**200K**	600K

Table 4. Provides insight into the maximum number of channels/memory for different architectures, considering 16-bit floats for activations. Notably, our model employs bottlenecks strategically, optimizing memory requirements and overall performance (Howard *et al.*, 2017).

MobileNet_V3
Howard *et al.*, (2019) MobileNet_V3 builds on the success of MobileNet_V2 by introducing several new features, including:

- Hardswish activation function: The hardswish activation function is a more efficient alternative to the ReLU activation function. It is a piecewise linear function that approximates the behavior of ReLU while being cheaper to compute.
- Squeeze-and-excitation (SE) modules: SE modules are a type of attention mechanism that is used to improve the feature representations of the network. They dynamically adjust the weight of each feature channel based on its importance.
- MISH activation function: The MISH activation function is a combination of the hardswish and sigmoid activation functions. It is designed to have the benefits of both activation functions, namely, being more efficient than ReLU and having a smooth output.

Mobilenet_V3 Architecture
The MobileNetV3 architecture as shown in Figure 2, is designed to optimize the accuracy-latency trade-off on mobile devices while introducing two new models, MobileNetV3-Large and MobileNetV3-Small, designed to target high and low-resource use cases and adapted and applied to tasks such as object detection and semantic

segmentation, achieving state-of-the-art results for mobile classification, detection, and segmentation (Howard *et al.*, 2019).

The architecture search for MobileNetV3 involves platform-aware NAS for block-wise search and NetAdapt for layer-wise search (Howard *et al.*, 2019). This process is repeated until the latency reaches its target, and then the new architecture is re-trained from scratch.

The efficient building blocks used for MobileNetV3 include depth-wise separable convolutions, linear bottlenecks, inverted residual structures, and lightweight attention modules based on squeeze and excitation. These building blocks are combined to create the most effective models, and layers are upgraded with modified swish nonlinearities. The architecture also uses a combination of these layers as building blocks to build the most effective models.

CNN Architecture

Convolutional Neural Networks (CNNs) are a type of deep learning architecture that is extensively used for image and video recognition, natural language processing, and other applications (Li *et al.*, 2016). CNNs are preferred in deep learning architecture because they utilize perceptrons for breaking down the gathered information. CNNs consists of a series of layers, including convolution layers, pooling layers, and normalization layers as shown in Figure 3.

- **Convolution Layer**: This layer involves the convolution of feature maps of the previous layer with a kernel. The output feature map of the convolution layer is computed using Eq. (1), which involves the convolution operator, sigmoid function, and input of the non-linear sigmoid function.

$$x_j^l = sigm(z_j^l),\ z_j^l = \sum_i x_j^{l-1} * k_{ij}^l + b_j^l \qquad (1)$$

- **Pooling Layer**: In this layer, sub-sampling of input features takes place to reduce the resolution of feature maps and increase the invariance of those features. The output feature map of the pooling layer is computed using Eq. (2), which involves the sub-sampling function for average pooling.

$$x_j^{l+1} = down(x_j^l) \qquad (2)$$

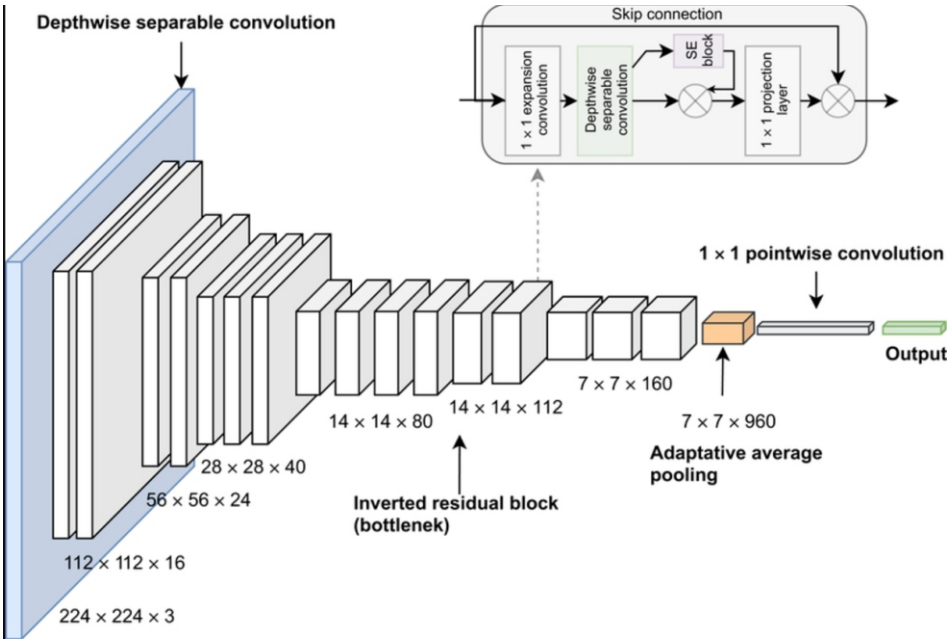

Figure 2: Architectural Diagram of MobileNet V3.

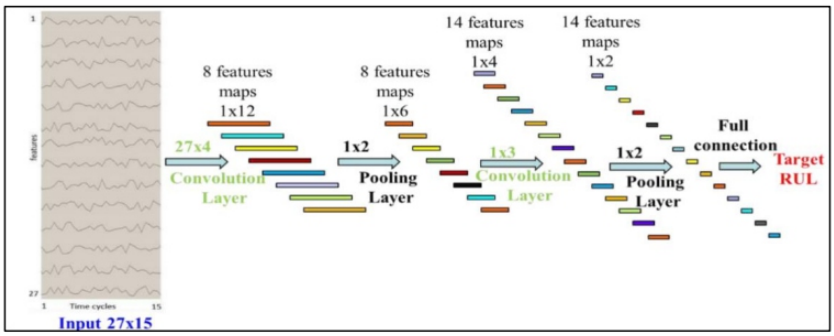

Figure 2: Architectural Diagram of MobileNet V3.

MATERIALS AND METHODS

Dataset Collection

To support our investigation in the realm of image collection, we accessed the PlantVillage dataset online at https://www.kaggle.com/datasets/emmarex/plantdisease in October 2023. This dataset, encompassing 3 crops namely, Potato, Pepper and Tomato, 15 directories of 20,639 high-quality JPEG images in a 5471x3648 pixel format. In the preprocessing phase, which includes noise removal and segmentation, the images are resized to 256x256 pixels (Gandhi *et al.*, 2018). Recognized as a prominent dataset for crop disease research, the PlantVillage dataset has been widely used, featuring images captured in a controlled laboratory environment, serving as training datasets of 20,639 plant leaf images, categorizing diseases within 15 sub-directories.

Our choice of collection focuses on disease-affected and healthy images, with examples such as Potato Blight, Tomato Spider, Tomato Mosaic, Pepper Bacteria spots, healthy spots, mosaic virus, septoria leaf spot, bacterial spot, early blight, late blight, septoria leaf spot, and spider mites, specifically in the context of potato, pepper and tomato imagery. This approach ensures that our research incorporates real-world scenarios and diverse environmental conditions, enhancing the robustness and applicability of our findings (Gandhi *et al.*, 2018).

The conceptual representation in ***Figure* 4** outlines the research approach for the classification and analysis of multi-crop leaf diseases. The initial step involves the collection of plant leaf disease images, which are then categorized. Various image processing techniques such as filtering, sharpening, grey-transforming and scaling the picture are employed in the preprocessing phase. Data enhancement methods were applied to enhance and prepare the dataset which led to the generation of new sample photos from the existing ones. Techniques like rotation, translation, and randomized transformation were utilized to expand the dataset's size. Subsequently, these augmented images serve as input for training the proposed model in the next stage.

The newly trained architectural models are then tested to predict outcomes for previously unseen images. Then the results from testing these models are juxtaposed, analyzed and documented.

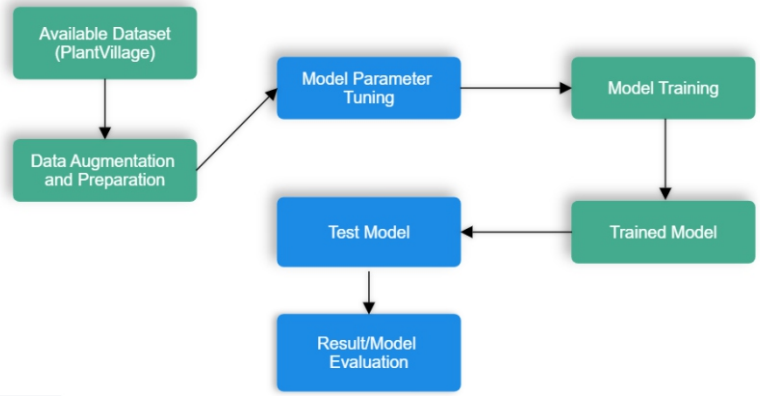

Figure 4. Research Approach Flow Diagram.

Data Augmentation

Data augmentation was done using specific augmentation parameters which include rotation, width shift, height shift, shear, zoom, horizontal flip, and fill mode which enhanced generalization, and decreases model overfitting, thereby improving model performance and aiding in the learning of robust characteristics by models, improving accuracy and resistance to noise and changes in the actual world.

Rationale for Choosing Models

An extensive review of existing literature and previous research studies informed the selection process, emphasizing models that have demonstrated success in similar contexts. Additionally, the decision to employ transfer learning and pre-trained models was motivated by the desire to leverage knowledge gained from large-scale datasets, enhancing the models' ability to generalize effectively to our dataset.

Model Building and Evaluation

In the case of the PlantVillage dataset being used, the images are standardized to a size of 224x224 pixels to train the Mobilenet_V2, Mobilenet_V3 and a custom-built CNN model. By modifying the pooling size, a smaller image seamlessly fit into the network. Proper preparation of the images ensured effective transfer learning with the multi-cropped image dataset.

A CNN_Model (Convolutional Neural Network) was trained using the Keras library in the following steps:

1. Model Initialization: A Sequential model is instantiated from Keras. The Sequential model allows for the creation of a linear stack of layers.
2. Input Layer and Convolutional Layers: The first layer added to the model is a 2D convolutional layer (`Conv2D`). This layer is responsible for learning spatial hierarchies from the input images. Additional convolutional layers are added with increasing complexity, employing the rectified linear unit (ReLU) activation function and batch normalization for improved training stability.
3. Max Pooling and Dropout: Max pooling layers was introduced to down-sample the spatial dimensions of the feature maps, reducing computation and preventing overfitting.
4. Flatten and Dense Layers: We employed the Flatten layer to flatten the 3D feature maps to 1D, enabling the connection to densely connected layers. Next, Dense layers (fully connected layers) are added to the model for classification.
5. Model Compilation: We compiled the model with an optimizer, loss function (binary cross-entropy for binary classification), and metrics (accuracy). Next, we displayed a summary of the model architecture and parameters.
6. Model Training: During training, the model is trained using the training data and evaluated on the validation data.
7. Evaluation and Saving: The model's performance is evaluated on the test data, and the trained model is saved to disk. The sequential construction of layers, activation functions, and optimization settings defines the architecture and behaviour of our cnn_model.

Platform Utilization: Kaggle

The experimental setup took advantage of Kaggle's platform for its computational resources and collaborative environment. By creating a dedicated notebook on Kaggle, the research harnessed the platform's capabilities for seamless model development, training, and evaluation. Kaggle's accessible computing power allowed for the efficient execution of complex deep learning algorithms, ensuring the scalability of the experiments. Kaggle is a cloud-based computing platform for data science and can be found at www.kaggle.com.

EXPERIMENT AND RESULTS

Experiment 1: Mobilenet_V2 Model

The first experiment was carried out on the Mobilenet_V2 model

Training Results

- Training and Validation Loss Curves

The Mobilenet_V2 model underwent an extensive training process to learn the intricate patterns

within the plant disease dataset. The training loss curves depict the model's convergence during the training phase. These curves illustrate the progression of the loss function over successive epochs as shown in Figure 5, providing insights into the model's ability to minimize errors and improve its predictive capabilities.

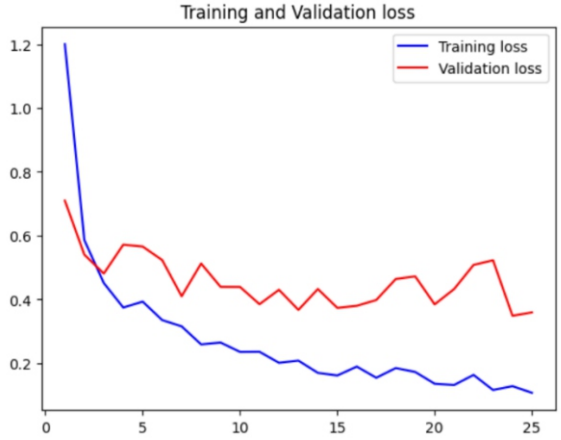

Figure 5. Graph showing training and Validation Loss Curves for experiment 1

Evaluation Results

Validation and Test Accuracy Trends

As shown in Figure 5 and Figure 6, the validation accuracy trends offer a glimpse into the model's generalization performance. Monitoring accuracy on a separate validation set during training helps identify potential overfitting or under-fitting issues. This section analyzes how the Mobilenet_V2 model's accuracy evolves on the validation set across training epochs shown in Table 5.

Table 5 below, summarily, shows and documents the training progress of our MobileNet_V2 model, demonstrating improvements in training and validation metrics over 25 epochs. The model is learning well and generalizing effectively to our image data.

Table 5: Epoch Result for Mobilenet_V2.

EPOCH	Time to run	Loss	Accuracy	Value Loss	Value Accuracy
1/25	52s 674ms/step	1.2002	0.6239	0.7096	0.7716
2/25	48s 660ms/step	0.5858	0.7986	0.5402	0.8105
3/25	48s 651ms/step	0.4512	0.8480	0.4809	0.8376
4/25	49s 675ms/step	0.3744	0.8716	0.5715	0.8003
5/25	49s 670ms/step	0.3929	0.8600	0.5655	0.8071
6/25	49s 666ms/step	0.3350	0.8832	0.5231	0.8190
7/25	50s 687ms/step	0.3154	0.8897	0.4098	0.8596
8/25	49s 667ms/step	0.2588	0.9141	0.4077	0.8666
9/25	49s 677ms/step	0.2646	0.9038	0.4392	0.8545
10/25	49s 675ms/step	0.2353	0.9158	0.4390	0.8494
11/25	51s 692ms/step	0.2357	0.9193	0.3848	0.8680
12/25	49s 673ms/step	0.2011	0.9326	0.4299	0.8443
13/25	48s 661ms/step	0.2079	0.9304	0.3670	0.8782
14/25	49s 670ms/step	0.1695	0.9420	0.4324	0.8596
15/25	49s 673ms/step	0.1613	0.9489	0.3732	0.8799
16/25	50s 688ms/step	0.1892	0.9373	0.3798	0.8579
17/25	50s 683ms/step	0.1541	0.9459	0.3980	0.8646
17/25	50s 683ms/step	0.1541	0.9459	0.3980	0.8646
18/25	49s 671ms/step	0.1846	0.9390	0.4637	0.8494
19/25	50s 678ms/step	0.1724	0.9403	0.4722	0.8511
21/25	49s 670ms/step	0.1315	0.9541	0.4323	0.8579
22/25	49s 668ms/step	0.1631	0.9429	0.5080	0.8376
23/25	50s 683ms/step	0.1157	0.9562	0.5225	0.8409
24/25	50s 683ms/step	0.1278	0.9575	0.3486	0.8866
25/25	49s 673ms/step	0.1070	0.9648	0.3591	0.8883

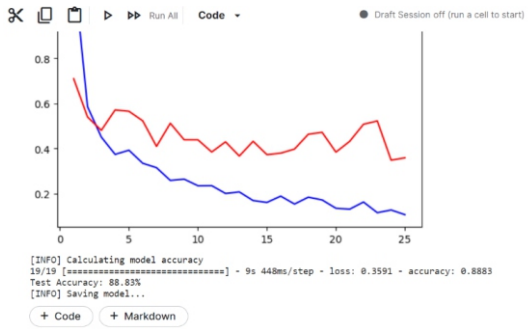

Figure 6. Graph showing validation and test accuracy for experiment 1.

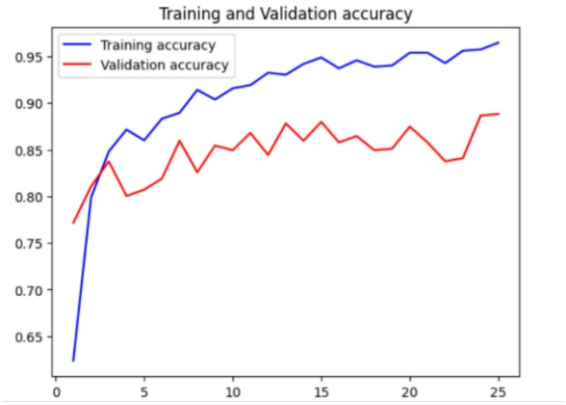

Figure 7. Graph showing training and validation accuracy for experiment 1.

Experiment 2: Mobilenet_V3 Model

Training Results

- Training Loss Curves

The training process of the Mobilenet_V3 model involved the optimization of model parameters to minimize the training loss. The training loss curves as shown in ***Figure 8***, depict the convergence of the model during this process, providing insights into its ability to learn and adapt to the complexities of the plant disease dataset.

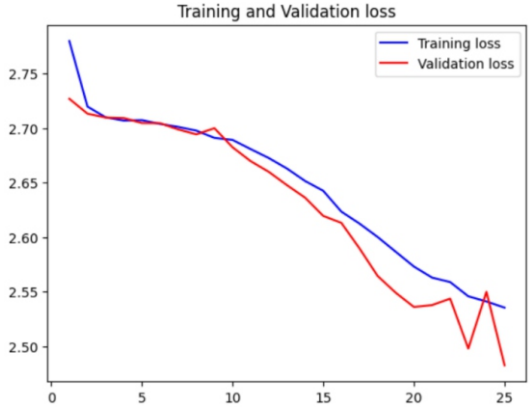

Figure 8. Graph showing Training and Validation Loss Curves for Experiment 2

- Test and Validation Accuracy Trends

Validation accuracy trends (as shown in ***Figure 9***) illustrate how well the Mobilenet_V3 model performs to data during the training phase. Monitoring these trends helps identify potential overfitting or under fitting issues, offering a deeper understanding of the model's robustness.

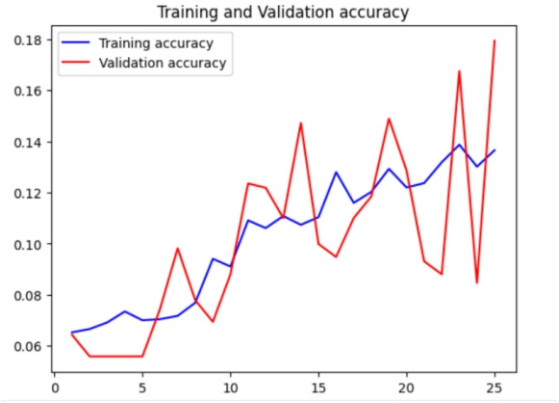

Figure 9. Training and Validation Accuracy for Experiment 2

The evaluation of the Mobilenet_V3 model on a previously unseen test dataset is summarized by the test accuracy metric (see ***Figure 9***). This provides a comprehensive measure of the model's performance in accurately classifying plant diseases in real-world scenarios.

In summary, **Table 6** documents the training progress of our MobileNet_V3 model, showcasing trends in training and validation metrics over 25 epochs.

Table 6: Epoch Result for Mobilenet_V3.

EPOCH	Time to Run	Loss	Accuracy	Value Loss	Value Accuracy
Epoch 1/25	46s 577ms/step	2.7798	0.0653	2.7269	0.0643
Epoch 2/25	41s 558ms/step	2.7197	0.0666	2.7132	0.0558
Epoch 3/25	39s 532ms/step	2.7099	0.0691	2.7097	0.0558
Epoch 4/25	39s 538ms/step	2.7069	0.0734	2.7092	0.0558
Epoch 5/25	39s 532ms/step	2.7073	0.0700	2.7046	0.0558
Epoch 6/25	41s 558ms/step	2.7038	0.0704	2.7045	0.0745
Epoch 7/25	40s 553ms/step	2.7011	0.0717	2.6988	0.0981
Epoch 8/25	40s 551ms/step	2.6977	0.0769	2.6942	0.0778
Epoch 9/25	39s 532ms/step	2.6910	0.0940	2.6999	0.0694
Epoch 10/25	41s 556ms/step	2.6891	0.0910	2.6824	0.0880
Epoch 11/25	41s 558ms/step	2.6807	0.1091	2.6696	0.1235
Epoch 12/25	40s 541ms/step	2.6725	0.1061	2.6598	0.1218
Epoch 13/25	40s 552ms/step	2.6629	0.1108	2.6476	0.1100
Epoch 14/25	40s 552ms/step	2.6514	0.1073	2.6361	0.1472
Epoch 15/25	41s 553ms/step	2.6425	0.1103	2.6194	0.0998
Epoch 16/25	39s 532ms/step	2.6233	0.1280	2.6130	0.0948
Epoch 17/25	40s 553ms/step	2.6124	0.1159	2.5896	0.1100
Epoch 18/25	41s 557ms/step	2.6001	0.1202	2.5643	0.1184
Epoch 19/25	40s 552ms/step	2.5865	0.1292	2.5489	0.1489
Epoch 20/25	39s 536ms/step	2.5729	0.1219	2.5358	0.1286
Epoch 21/25	40s 552ms/step	2.5628	0.1237	2.5376	0.0931
Epoch 22/25	41s 557ms/step	2.5587	0.1318	2.5435	0.0880
Epoch 23/25	40s 545ms/step	2.5456	0.1387	2.4977	0.1675
Epoch 24/25	41s 558ms/step	2.5409	0.1301	2.5498	0.0846
Epoch 25/25	39s 539ms/step	2.5352	0.1365	2.4823	0.1794

Experiment 3: CNN Model

Training Results

- Training Loss Curves

Similar to the pre-trained models, the cnn_model underwent training, and the training loss curves illustrate the model's learning dynamics (See ***Figure 10***). These curves provide insights into how well our trained CNN model adapts to the unique characteristics of the plant disease dataset.

- Validation Accuracy Trends

The validation accuracy trends for our cnn_model showcase its ability to generalize to new instances during the training phase. Monitoring these trends is crucial for understanding the model's capacity to recognize patterns in the data.

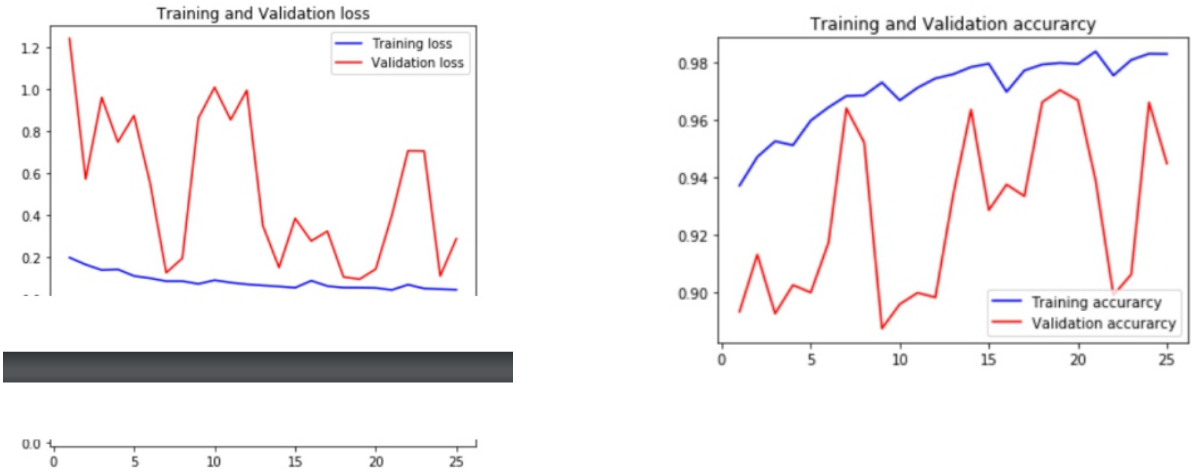

Figure 10. Shows both training and validation loss and accuracy curves for experiment 3.

Evaluation Results

- Test Accuracy

The evaluation of our cnn_model on the test dataset is summarized by the test accuracy metric. This metric reflects the model's overall performance in identifying plant diseases and provides a basis for comparison with other models.

Comparative Analysis

- Performance Metrics across Models

In this section, we tabulate the performance metrics across Models in *Table 5*, presenting a comparative analysis of performance metrics, including accuracy, precision, recall, and F1 score, across the Mobilenet_V2, Mobilenet_V3, and CNN_model. Examining these metrics collectively provides a holistic view of each model's capabilities.

Table 7. Showing Comparison of performance between Mobilenet_V2, Mobilenet_V3 and our CNN_model

Model	Test Accuracy	precision	recall	f1 score
Mobilenet_V2	88.83%	0.87	0.89	0.88
Mobilenet_V3	17.94%	0.18	0.18	0.18
CNN_Model	94.48%	0.95	0.94	0.94

DISCUSSION

The custom-built CNN model achieved the highest test accuracy of 94.48%, surpassing MobileNet_V2 (88.83%) and MobileNet_V3 (17.94%). Its superior precision (0.95), recall (0.94), and F1-score (0.94) align with literature emphasizing CNNs' strength in image-based tasks (Xie *et al.*, 2020). This performance also exceeds benchmarks from models like VGGNet (91.83%, Xie *et al.*, 2020) and other standard classifiers, highlighting the effectiveness of tailored architectures for specific datasets.

MobileNet_V2 demonstrated competitive performance, reflecting its efficiency in simpler use cases, consistent with findings by Howard *et al.* (2017). However, MobileNet_V3's low accuracy (17.94%) contrasts sharply with its expected results from other studies (Howard *et al.*, 2019). This underperformance likely stems from challenges in fine-tuning and adapting its advanced features, such as SE modules and Hardswish, to the dataset. These results show the importance of custom models in achieving superior performance for domain-specific tasks. While MobileNet architectures are efficient for general-purpose mobile applications, the CNN model's adaptability and robustness make it more

suitable for the agricultural context studied here. Further exploration of tailored and hybrid models could enhance future performance and practical applicability.

CONCLUSION

Summary of Findings

The findings of the research are summarized, providing a comprehensive overview of the outcomes from the experiments conducted on Mobilenet_V2, Mobilenet_V3, and our CNN model for plant disease detection. The key takeaways from the training and evaluation results have been outlined, emphasizing the performance variations among the models. Our CNN model emerged as the most effective, achieving a test accuracy of 94.483%, outperforming both Mobilenet_V2 and Mobilenet_V3.

Future Research Directions

As we conclude this research journey, several promising avenues for future exploration emerge:

Model Ensemble Techniques: Investigate the integration of ensemble techniques to combine the strengths of multiple models, potentially enhancing overall performance and robustness in plant disease detection.

Explainability and Trustworthiness: Delve deeper into the development of explainable AI models to enhance the interpretability and transparency of complex models. Building trust in decision-making processes is critical for the adoption of machine learning in agriculture.

Continuous Dataset Evolution: Explore strategies for continuous dataset augmentation and evolution to ensure that machine learning models remain adaptive to evolving disease patterns and environmental conditions in real-world agricultural settings.

***Collaboration with Domain Experts*:** Foster collaboration between machine learning practitioners and domain experts, including plant pathologists and agronomists, to refine models based on practical insights and ensure their alignment with the needs of the agricultural community.

ACKNOWLEDGEMENT & FUNDING

Support for implementation of project activities was made possible by the Research Grant (109705-001/002) by the Responsible Artificial Intelligence Network for Climate Action in Africa (RAINCA) consortium made up of WASCAL, RUFORUM and AKADEMIYA2063 provided by IDRC.

CONFLICT OF INTEREST

The authors declare that no competing conflict of interest exists.

AUTHORS' CONTRIBUTIONS

N.C.B.: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Validation, Visualization, writing – original draft.

Y-B.C.: Conceptualization, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualization.

REFERENCES

Ai, Y., Sun, C., Tie, J., & Cai, X. 2020. Research on recognition model of crop diseases and insect pests based on deep learning in harsh environments. *IEEE Access* 8, 171686–171693. doi: 10.1109/access.2020.3025325

Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M. and Alrahamneh, Z., 2011. Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17(1), pp.31-38. doi.org/10.18280/ts.410337

Ang, K.L.M., Seng, J.K.P., 2021. Big data and machine learning with hyperspectral information in agriculture. IEEE Access 9, 36699–36718. doi: 10.1109/ACCESS. 2021.3051196

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A. and Stefanovic, D., 2019. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry*, 11(7), p.939. doi: 10.3390/sym11070939

Chen, C.J., Huang, Y.Y., Li, Y.S., Chen, Y.C., Chang, C.Y. and Huang, Y.M., 2021. Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access*, 9, pp.21986-21997. doi: 10.1109/ACCESS.2021.3056082

Chen, L., Lin, S., Lu, X., Cao, D., Wu, H., Guo, C., ... & Wang, F. Y. (2021). Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *22*(6), 3234-3246. doi: 10.1109/TITS.2020.2993926

Chirasani, S. K. R., Prabakaran, T., Fairooz, S., Munaswamy, P., Ashok, M., Sravanthi, G., ... & Rajeswaran, N. (2024). Edge Architecture for High-Accuracy Disease Identification in Apple Plants Using Transfer Learning Approach. *Traitement du Signal*, 41(3). doi: 10.18280/ts.410337

Gandhi R, S. Nimbalkar, N. Yelamanchili and S. Ponkshe, "Plant disease detection using CNNs and GANs as an augmentative approach," *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, Bangkok, Thailand, 2018, pp. 1-5. doi: 10.1109/ICIRD.2018.8376321

Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. and Le, Q.V., 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1314-1324. doi: 10.48550/arXiv.1905.02244

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Li, X., Zhang, Y., Marsic, I., Sarcevic, A. and Burd, R.S., 2016, November. Deep learning for rfid-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 164-175. doi: 10.1145/2994551.2994569

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L.R. and Davis, C.E., 2015. Advanced methods of plant disease detection. A review. *Agronomy for Sustainable Development*, *35*, 1-25. doi: 10.1007/s13593-014-0246-1

Massi, I.E., Mostafa, Y.E.-s., Yassa, E., Mammass, D., 2020. Combination of multiple classifiers for automatic recognition of diseases and damages on plant leaves. Signal Image Video Process. doi: 10.1007/s11760-020-01797-y

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510-4520. doi: 10.1109/CVPR.2018.00474

Shiferaw, B., Smale, M., Braun, H.J., Duveiller, E., Reynolds, M. and Muricho, G., 2013. Crops that feed the world 10. Past successes and future challenges to the role played by wheat in global food security. *Food Security*, 5, 291-317. doi: 10.1007/s12571-013-0263-y

Singh, A., Ganapathysubramanian, B., Singh, A.K. and Sarkar, S., 2016. Machine learning for high-throughput stress phenotyping in plants. *Trends in plant science*, *21*(2), 110-124. doi: 10.1016/j.tplants.2015.10.015

Sun, J., Yang, Y., He, X., Xiaohong, W., 2020. Northern maize leaf blight detection under complex field environment based on deep learning. *IEEE Access* 8, 33679–33688. doi: 10.1109/ACCESS.2020.2973658

Wei, W.U., YANG, T.L., Rui, L.I., Chen, C.H.E.N., Tao, L.I.U., Kai, Z.H.O.U., SUN, C.M., LI, C.Y., ZHU, X.K. and GUO, W.S., 2020. Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales. *Journal of Integrative Agriculture*, *19*(8), 1998-2008. doi: 10.1016/S2095-3119(19)62803-0

Xie, X., Ma, Y., Liu, B., He, J., Li, S. and Wang, H., 2020. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Frontiers in plant science*, *11*, 751. doi: 10.3389/fpls.2020.00751

Zhou, C., Zhou, S., Xing, J. and Song, J., 2021. Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access*, *9*, 28822-2883. doi: 10.1109/ACCESS.2021.3058947