Leveraging MobileNetV3 for In-Field Tomato Disease Detection in Malawi via CNN

Lindizgani K. Ndovie, Emmanuel Masabo

Abstract— **Malawi's economy heavily depends on agriculture, including both commercial and subsistence farming. Smallholder and small-medium enterprises leading the production of tomatoes in Malawi cannot satisfy local demand due to problems such as pests, diseases, unstable markets, and high costs. Many farmers lack the expertise to effectively manage these threats. To address the problem of tomato leaf disease identification, this research aimed to develop an automated system for tomato leaf disease detection by utilizing data augmentation techniques, MobileNetV3, and Convolutional Neural Network algorithms. We trained models on secondary data collected from the public PlantVillage dataset and tested the resultant classifiers on primary data of local farm images. The experimental results demonstrate that both models tested better on the PlantVillage dataset. Additionally, with an accuracy of 92.59% and a loss of 0.2805, the pre-trained MobileNetV3 model conventionally performs better than a CNN model. However, when tested on the primary field dataset, the models did not meet expectations for generalization, with the pre-trained MobileNetV3 achieving an accuracy of 9.2%, and loss of 12.91 and the CNN achieving an accuracy of 10.14%, and loss of 8.11. The experiments aided in showing that the models trained on the PlantVillage dataset are not as effective when applied in real-world scenarios. Further improvements are needed to enhance the models' generalization in real-world scenarios.**

Index Terms— **Convolutional Neural Network, Deep learning, Image augmentation, MobileNetV3, Tomato leaf disease detection**

I. Introduction

Tomatoes play a vital role in the agricultural sector of Malawi, serving as a significant crop for food consumption and income generation [1]–[4]. However, small-scale and semi-commercial tomato growers face various challenges, including pests, diseases, marketing difficulties, and high input prices, resulting in inadequate supply to meet local demand [1], [4], [5]. According to Eviness et al. [1], tomato producers usually cultivate during the dry season due to the reduced prevalence of diseases and pests. This concentration of tomato production in the dry season leads to a surge in supply and creates a spillover effect that results in lower prices. To address these challenges and improve tomato production, it is crucial to adopt modern solutions, particularly in pest and disease management.

The current state of research in the field highlights the limitations of traditional diagnostic methods, which are often costly, time-consuming, and subjective. Visual assessment introduces bias and errors, making accurate disease diagnosis challenging. A study conducted in major tomato-growing districts in Malawi highlighted the lack of formal education among local tomato farmers, which limits their ability to identify and manage detrimental diseases and select appropriate pesticides [1]. Most local farmers currently utilize traditional methods that have been gained through experience and oral knowledge passed down through generations. This knowledge gap necessitates the development of automated approaches that can rapidly and accurately detect tomato diseases, providing a convenient and accessible solution.

This paper aligns with the growing interest in developing automated approaches for disease detection using deep-learning techniques. Different research findings and approaches exist within the field regarding the effectiveness of data augmentation techniques in improving model performance, the superiority of pre-trained models over convolutional neural networks (CNN), and the challenges associated with applying such models in real-world scenarios. These varying perspectives emphasize the need for further investigation and evaluation to advance our understanding of automated disease detection methods for tomatoes.

The purpose of this study was to train MobileNetV3 and CNN models on a public dataset of tomato leaf diseases and to evaluate the resultant models on real field images collected in Malawian farms. By utilizing data augmentation techniques, MobileNetV3, and CNN algorithms, the study aimed to provide an accurate, reliable, and efficient automated method for identifying and managing tomato diseases. Ultimately, this automated method could be integrated into a mobile platform for the benefit of local farmers in Malawi's agriculture sector.

II. Literature Review

Diagnosing tomato plant diseases traditionally involves visual inspections and manual analyses by plant pathologists, who rely on symptoms like leaf discoloration and spots for identification [6]. These methods, although grounded in expertise, are hindered by their time-consuming nature, high costs, and the potential for human error. Diagnostic techniques, including microscopic examination and the isolation of

This work was supported by the African Centre of Excellence in Data Science (ACE-DS), University of Rwanda.

Lindizgani Kingstone Ndovie is with the African Centre of Excellence in Data Science, University of Rwanda (e-mail: lindizganindovie@gmail.com).

Emmanuel Masabo is with the African Centre of Excellence in Data Science (ACE-DS) and the African Centre of Excellence in Internet of Things (ACE-IoT), University of Rwanda (e-mail: masabem@gmail.com).

pathogens, require specialized skills and are subject to variability in interpretation [7]. The limitations of these approaches underscore the necessity for automated, reliable disease detection systems to aid growers and pathologists.

A. Deep Learning in Disease Detection

The application of Convolutional Neural Networks (CNNs) represents a significant advancement in plant leaf disease detection, offering a more efficient alternative to traditional algorithms [8], [9], [10].

CNNs streamline the processing of images through layers that automatically extract and learn features, significantly reducing the need for manual pre-processing. Despite their advantages in handling complex image data, CNNs need substantial computational resources and training time, which poses challenges to their widespread adoption [11], [12].

To resolve the problem of requiring a lot of processing time and power, pre-trained neural networks are used. These are models that have been trained on a large dataset and are designed to be efficient and powerful requiring fewer parameters and computations for their specific task. This form of transfer learning allows CNNs to quickly learn from existing models without having to start from scratch. An example of a pre-trained architecture is the MobileNetV3. This is a pre-trained convolutional neural network that was designed for mobile applications by Google in 2019 [13].

Literature suggests that MobileNetV3 architecture is a strong choice for applications that require high accuracy, as it balances efficient computation with competitive performance through various design principles. The main difference between a CNN model and a MobileNetV3 model lies in their architectural design. A CNN has multiple convolutional layers, pooling layers, and fully connected layers without much tweaking for optimization. A MobileNetV3 model, on the other hand, has its base model as a CNN. However, it has additional optimization techniques to decrease the number of parameters and the processing cost during training. Furthermore, a CNN model might possess more parameters and computational requirements compared to MobileNetV3, leading to it potentially being slower and requiring more memory to process. MobileNetV3, alternatively, is known for its lightweight and efficient design, making it suitable for resource-constrained environments such as mobile devices, vis-à-vis the name of the model. It is faster and more memory-efficient as compared to more complex CNN architectures.

B. Role of Data Augmentation in Improving Detection

Data augmentation is a method for expanding a dataset artificially by making transformed copies of a dataset from existing data [11], [14]. Data augmentation is usually used as a regularization technique to reduce the overfitting of a model. While several techniques exist for image augmentation, the method of geometric transformations is the most commonly used. Images are shifted, rotated, resized (among other processes) and added to the dataset during this process. Color transformations, such as adjusting brightness, contrast, and hue, can simulate different lighting conditions and improve the model's ability to handle variations in image quality.

Since leaf disease symptoms can vary in terms of size, color, shape, and severity, it is challenging to develop robust detection models that can accurately identify and classify different diseases [15]. Data augmentation techniques offer a promising approach to enhance the performance of disease detection models by diversifying the training dataset and increasing its size [16].

C. Available Datasets for Tomato Leaf Diseases

The use of comprehensive datasets such as PlantVillage and PlantDoc plays a critical role in training and validating disease detection algorithms [6], [17]. These datasets provide a diverse array of images that are instrumental in developing models capable of recognizing a broad spectrum of plant diseases. However, challenges remain in ensuring these datasets accurately reflect the variety of disease symptoms across different environments, highlighting the need for ongoing efforts to expand and diversify the data available for research purposes.

D. Related Work

This subsection provides insights into various studies that have contributed significantly to advancements in the field of tomato plant disease detection. These studies highlight the diversity in methodologies, ranging from the optimization of CNN architectures to the exploration of hybrid models and the utilization of both public and private datasets to achieve improved disease classification accuracy.

A study by Agarwal, Singh, et al. [18] focused on training models using the PlantVillage dataset, with a particular emphasis on expanding the training set through data augmentation for classes with insufficient images. This approach aimed to mitigate bias by ensuring a balanced representation of classes. The research compared the performance of a custom CNN architecture against pre-trained models like VGG16, InceptionV3, and MobileNet, concluding that the custom CNN approach, despite its simplicity, was less effective for the specific task of tomato leaf disease classification compared to the more complex pre-trained models.

Nandhini & Ashokkumar [19] explored the optimization of VGG16 and Inception V3 models for tomato leaf disease detection, achieving remarkably high accuracies. Their study stands out for its methodological rigor in fine-tuning models to suit the specificities of plant disease detection, showcasing the potential of optimized deep-learning models in agricultural applications.

The work of Liu & Wang [20] delves into the development of hybrid models that combine MobileNetV2 and YOLOv3 for enhanced performance. Their approach, tailored for early detection of grey leaf spot disease, underscores the innovative use of hybrid learning to address challenges in disease detection under varying environmental conditions. This study is particularly notable for its application to images collected in natural field settings, providing valuable insights into model performance in real-world scenarios.

Chen et al. [21] leveraged a private dataset to investigate the efficacy of a modified ResNet50 model in identifying tomato leaf diseases. Their research contributes to the understanding of how image enhancement techniques, such as Binary Wavelet Transformations, can be employed to improve model accuracy, offering a promising direction for future studies.

Gonzalez-Huitron et al. [22] employed the PlantVillage dataset, enhanced through data augmentation, to conduct a comparative analysis of transfer learning models including MobileNetV2, MobileNetV3, NasNetMobile, and Xception. Their findings, which favored the MobileNetV3 model for its superior accuracy and efficiency, highlight the importance of selecting the appropriate model architecture for specific detection tasks.

E. Research Gap

Despite these advancements, a consistent research gap is the limited evaluation of these models in practical, real-field conditions, particularly for tomato leaf diseases. Most studies rely on laboratory or controlled datasets, which may not fully capture the complexity and variability of diseases in natural environments. This gap underscores the necessity for further research that tests and validates the effectiveness of deep learning models on diverse, real-world datasets, bridging the gap between theoretical research and practical application in agriculture.

III. Methodology

This section describes the proposed methodology shown in Fig. 1 below.

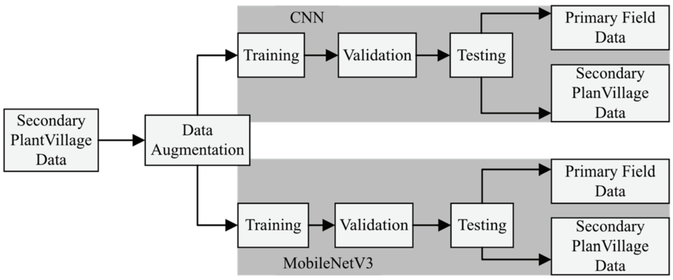

Fig. 1. Flowchart showing the proposed methodology for tomato leaf disease detection using CNN and MobileNetV3.

A. Datasets Description

Two sets of data were used for this research. The primary data collection focused on local tomato farms in Lilongwe, Malawi. Images were captured using two standard smartphones, a Samsung A12 and a Xiaomi Redmi Note 10 Lite, which have quad-camera systems with varying specifications. The tomato leaf photos were taken from the upper surface of the leaves, with the crops' leaves being plucked and placed against a white backdrop in natural light. The images were then edited to show only the white background.

In addition to the primary data, the research also utilized the PlantVillage dataset [23], an open-source collection of images from experimental research stations at Land Grant Universities in the United States. The dataset consists of images captured using a Sony DSC-Rx100/13 digital camera, with diseased leaves placed against a grey or black backdrop. All photos were taken in natural light and subsequently edited by removing the background and orienting the leaves upright.

The primary dataset included a total of 217 images of tomato leaves, covering eight classes of diseases. However, two diseases (mosaic virus and spider mites) were not found, and were, therefore, not included in the dataset. The secondary PlantVillage dataset, on the other hand, comprised 55,000 leaf images from various crops, including tomatoes. For this research, a subset of the dataset containing ten classes of tomato leaf diseases (including a healthy class) was selected.

B. Data Pre-processing

Three subsets of data were used: training, validation, and testing. We did not divide the primary dataset which was used to test the models and we used it without alteration. We divided the secondary PlantVillage dataset into two subsets, namely, training (70%) and validation (30%). The training subset had 11,208 images, the validation dataset had 4,803 images, and the testing dataset had 217 images. The distribution of images across the classes in both datasets was unbalanced. The number of images found in each class of both datasets have been illustrated in Table I and some sample images of the tomato leaves have been shown in Fig. 2 and Fig. 3.

As a prerequisite for pre-processing, we scaled down all the image pixels by the factor 255 to be in the range [0,1] using the 'rescale' Keras pre-processing layer. We employed this technique to expedite and simplify computational demands during training.

We used data augmentation to improve diversity in the training subset by generating new images from the images available. To enhance the training set's diversity, we applied the following minor transformations to the original images:

- Geometric transformation: We randomly flipped, rotated, and zoomed the images.
- Noise addition: We introduced random noise to the images using a Gaussian Noise pre-processing layer with a standard deviation of 0.1.
- Color alteration: We randomly adjusted the brightness and contrast of the images.

Additionally, we used Keras pre-processing layers for the online augmentation of the dataset. This process of online data augmentation occurs when transformations are performed randomly on a mini-batch of the data during training [24]. This approach is particularly beneficial for large datasets as it avoids creating a significantly large dataset and mitigates potential overfitting issues in the model.

Table I: Number of Images Available in the Datasets

Class	Primary Dataset (Testing Data)	Secondary Dataset (Training and Validation Data)	
		Training Data	Validation Data
Healthy	49	1484	643
Bacterial Spot	25	690	310

Early Blight	11	1346	563
Late Blight	16	672	280
Leaf Mold	49	1203	568
Mosaic Virus	0	1180	496
Septoria Leaf Spot	27	975	429
Spider Mites	0	2282	926
Target Spot	19	265	108
Yellow Leaf Curl Virus	21	1111	480
Total	217	11208	4803

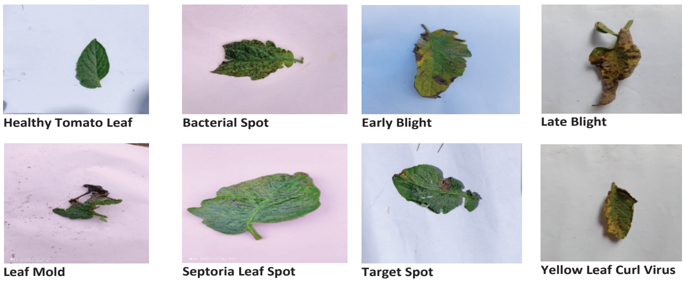

Fig. 2. Sample images of tomato leaves from the primary dataset.

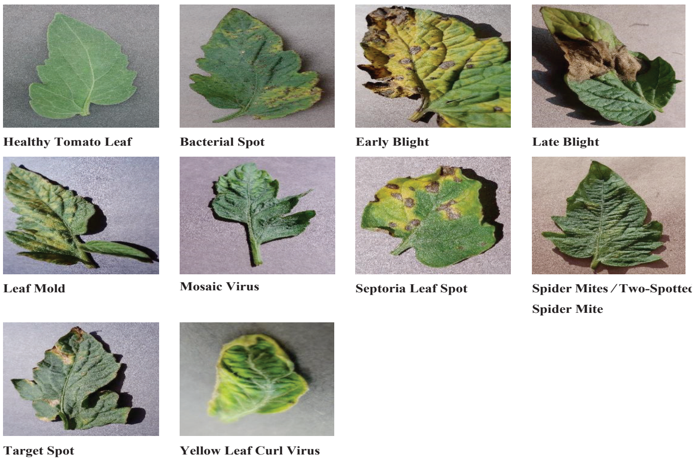

Fig. 3. Sample images of healthy and different unhealthy tomato leaves from the secondary dataset.

C. Data Analysis Procedures

1) Convolutional Neural Network (CNN)

A CNN is an artificial neural network that consists of three main layers: the convolutional layer, the pooling layer, and the fully connected layer. Each layer plays a unique role in processing images and extracting meaningful information. The convolutional layer utilizes learnable filters to create feature maps by convolving the image data. The most popular layers are max pooling and global pooling, which are used to minimize the feature map sizes and extract the most important characteristics. They can also be used to make the feature map more robust to small deformations or shifts in the input data. Finally, the fully connected layer combines the features extracted from the earlier layers to generate an output. In addition to these layers, CNNs often contain activation layers, normalization layers, and other components described under the hyperparameters section.

During this study, we trained a CNN algorithm on the secondary PlantVillage dataset (augmented). The architecture consists of 4 series of layers and is illustrated in Table II. The first series of layers consists of 2 convolutional layers with 32 filters of size 3x3, a max pooling layer with a pool size of 2x2, and a dropout layer with a rate of 0.5. It applies the ReLU activation function and expects input images of size 224x224x3. The second series of layers consists of 2 convolutional layers with 64 filters of a 3x3 size, a max pooling layer with a pool of 2x2 size, and a dropout layer with a rate of 0.5. The third series of layers consists of 2 convolutional layers with 128 filters of size 3x3, a max pooling layer with a pool size of 2x2, and a dropout layer with a rate of 0.5. The output of the previous layer is flattened into a one-dimensional vector in the fourth series of layers, and a dense layer with 10 nodes and "softmax" activation is used to predict the probability distribution of the input image over the 10 classes. This model's loss function represents categorical cross-entropy and uses the "Adam" optimizer with a learning rate of 0.0001.

Table II: Architectures of the CNN Model

Layer	Output Shape	Number of parameters
conv2d (Conv2D)	(None, 224, 224, 32)	896
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18496
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
dropout_1 (Dropout)	(None, 56, 56, 64)	0
conv2d_4 (Conv2D)	(None, 56, 56, 128)	73856
conv2d_5 (Conv2D)	(None, 56, 56, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_2 (Dropout)	(None, 28, 28, 128)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 10)	1003530
Total params: 1,290,538		
Trainable params: 1,290,538		
Non-trainable params: 0		

2) MobileNetV3

The MobileNetV3 architecture incorporates three elements designed to optimize performance and efficiency: bottleneck layers, inverted residual blocks, and linear bottlenecks. Bottleneck Layers reduce channel count while preserving activation information. Inverted Residual Blocks combine pointwise and depth-wise convolutions with ReLU activations to lower computational demands. Linear Bottlenecks eliminate non-linearities in narrow layers, fostering a compact, powerful network.

MobileNetV3 'Small' variant is chosen for its suitability in mobile applications, with transfer learning applied by freezing the pre-trained weights. A dropout layer (rate of 0.2) and a softmax-activated dense layer are appended for classification purposes.

Training occurs over several epochs, with batches undergoing forward and backward passes through MobileNetV3's base and trainable layers, employing categorical cross-entropy for loss calculation and Adam optimizer for weight updates.

Model performance is validated (accuracy, precision, recall, F1-score) with early stopping to mitigate overfitting, followed by model saving for future inference tasks.

The architecture is configured with an input layer, MobileNetV3 Small feature extractor, dropout layer (0.2 rate, L2 regularization of 0.0001), and a softmax-activated dense layer. The "Adam" optimizer and a categorical cross-entropy loss function facilitate the training process, with a learning rate of 0.005 optimizing the balance between speed and convergence.

The trained model is prepared for deployment in new disease detection applications. Incoming images undergo pre-processing to align with training specifications. Images are processed through the model, with softmax outputs dictating the probability distribution across classes, determining the predicted disease based on the highest probability.

Table III: Architectures of the MobileNetV3 Model

Layer	Output Shape	Number of parameters
keras_layer (KerasLayer)	(None, 1024)	1 026552
dropout_3 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 10)	10250
Total params: 1,036,802		
Trainable params: 10,250		
Non-trainable params: 1,026,552		

D. Hyper-parameters

Hyper-parameters are crucial variables that govern the training of a CNN and significantly impact its performance and accuracy. During the training of both the CNN model and the pre-trained MobileNetV3 model, a variety of hyper-parameters were fine-tuned (as presented in Table IV). Neurons, which represent functions that take multiple inputs and produce a single output, were fine-tuned to enhance the model's capabilities. The number of layers was crucial in influencing the complexity of the model. Additionally, the learning rate contributed significantly to determining the speed at which the model adjusted its weights and biases during training. Moreover, the batch size, referring to the number of samples processed before updating the model, played a vital role in affecting both the training time and the accuracy gains achieved.

We also carefully considered the epochs, representing the number of times the entire dataset was fed into the model during training. We meticulously controlled the steps per epoch, as they determined the number of times the model was trained on a given dataset before moving to the next epoch. To ensure an accurate evaluation, we employed validation steps to separate the training data from the validation data, enabling the assessment of the model's performance on unseen data. The "Adam" optimizer, known for its adaptability, was essential in modifying neural network parameters such as learning rates and weights. Activation functions, including ReLU and Softmax, introduced non-linearity and enabled effective multi-class classification.

Furthermore, to prevent overfitting, we strategically implemented dropout by randomly omitting neuron connections. Additionally, we employed early stopping as a technique to terminate training if the validation error exceeded a specified threshold; this approach helped us prevent overfitting the model. Lastly, the implementation of max pooling significantly reduced the computational workload by replacing multiple neurons with the maximum value among them, resulting in a faster training process and a reduction in network size.

Table IV: Hyper-parameters Used During Training

Parameter	CNN	MobileNetV3
Number of Epochs	24	26
Training time	172 seconds	118 seconds
Batch size	64	64
Optimizer	Adam	Adam
Learning Rate	0.0001	0.005
Dropout	0.5	0.2
Activation Function	Softmax	Softmax

E. Evaluation Metrics

We utilized various evaluation metrics to monitor the performance of the models during training. One of the key metrics used was the confusion matrix, which provided insights into the model's performance for each class in the dataset. We calculated true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) from the confusion matrix, allowing for a comprehensive analysis of the model's predictions.

Accuracy, another important metric, measured the proportion of correctly labeled images out of the total number of samples. However, we acknowledged that the dataset used in this research was unbalanced, which could introduce bias in the accuracy calculation towards the class that has the most samples, which is the yellow leaf curl virus. We used categorical cross-entropy loss in conjunction with accuracy to assess the model's performance, aiming to minimize the average number of mistakes made during predictions.

Precision, a metric focusing on positive labels, measured the probability of correctly identifying positive cases. Recall, also known as sensitivity, gauged the model's ability to correctly identify actual positive cases. F1-score, on the other hand, provided a balanced evaluation of the model's performance by combining precision and recall. False negative rate (FNR) measured the rate at which positive values were misclassified as negative, highlighting the model's performance in avoiding false negatives.

IV. RESULTS

This chapter presents the results of all the processes and procedures described in the research methodology. It presents the results after training both the CNN and the MobileNetV3 model in terms of the accuracy, loss, precision, recall, F1 score and confusion matrix. We interpret that a model is a good classifier when its validation accuracy is greater than its testing accuracy, when its validation loss is greater than its testing loss, or when the difference between its validation loss and training loss is approximately 0.

A. Convolutional Neural Network

During training, the model ran for 24 epochs for 172 seconds and Fig. 3 and Fig. 4 show the performance of the model during training and validation. The training accuracy was 0.8904 and the training loss was 0.3322 while the validation accuracy was 0.8371 and the validation loss was 0.4800. As represented, the validation accuracy was lower than the training accuracy, which interprets into two conclusions: that this model is not reliable at generalization and that it may not be a suitable classifier. Accuracy and loss were interpreted with caution because the dataset is heavily imbalanced. Nonetheless, the high accuracy gives a good indication of the model's performance.

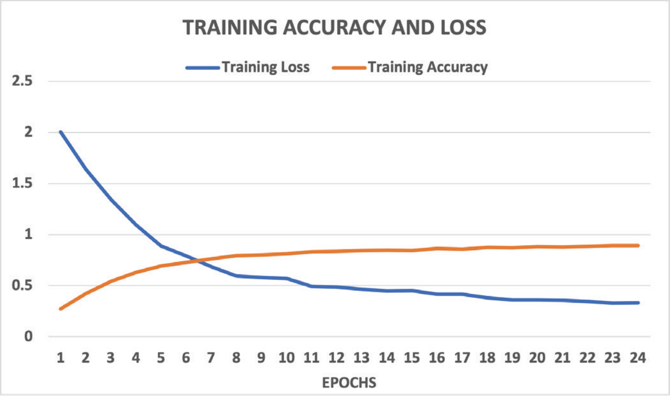

Fig. 4. Graphs of accuracy vs loss during training for CNN

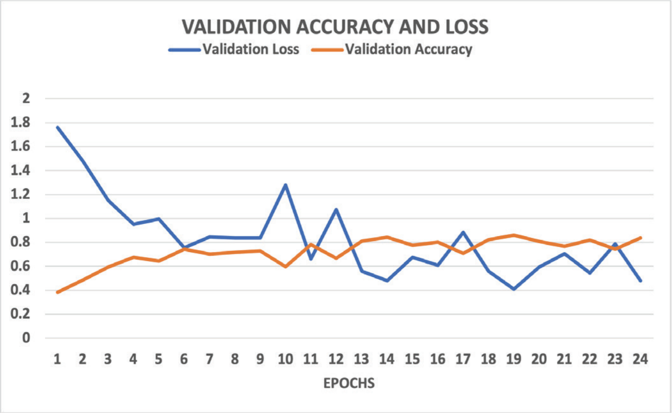

Fig. 5. Graphs of accuracy vs loss during validation for CNN

During testing using the secondary PlantVillage dataset, this model achieved an accuracy of 0.8603 and a loss of 0.4135. From a deeper evaluation using metrics like F1 Score, Precision, and Recall and the confusion matrix, as in Table V and Fig. 5 it becomes apparent that the CNN model trained on augmented data exhibits certain strengths and limitations. "Yellow Leaf Curl Virus" achieved the highest observed F1 score at 22.1%. This underlines that the model is most adept at detecting this disease among others. However, given that this score is still significantly below the optimal, there is undeniably room for further optimization. On the other end of the spectrum, "Leaf Mold" comes in with the lowest F1 score at 8.53%, signaling that the model faces challenges in reliably identifying this disease. In essence, most of the classes seem to put the model to the test, as evidenced by all the scores falling below 25%. Such scores might be insufficient for real-world applications, necessitating model refinements.

Table V: Evaluation Metrics When Testing the CNN On Secondary PlantVillage Dataset

Class	F1 Score	Precision	Recall
Healthy	14.51%	13.88%	15.21%
Bacterial Spot	17.36%	18.34%	16.49%
Early Blight	9.77%	11.71%	8.39%
Late Blight	16.14%	17.10%	15.28%
Leaf Mold	8.53%	10.58%	7.14%
Mosaic Virus	9.80%	10.42%	9.26%
Septoria Leaf Spot	17.12%	16.22%	18.13%
Spider Mites	16.57%	14.37%	19.56%
Target Spot	14.86%	13.99%	15.85%
Yellow Leaf Curl Virus	22.10%	22.51%	21.71%
Macro Average	15.00%	15.00%	15.00%
Weighted Average	16.00%	16.00%	16.00%

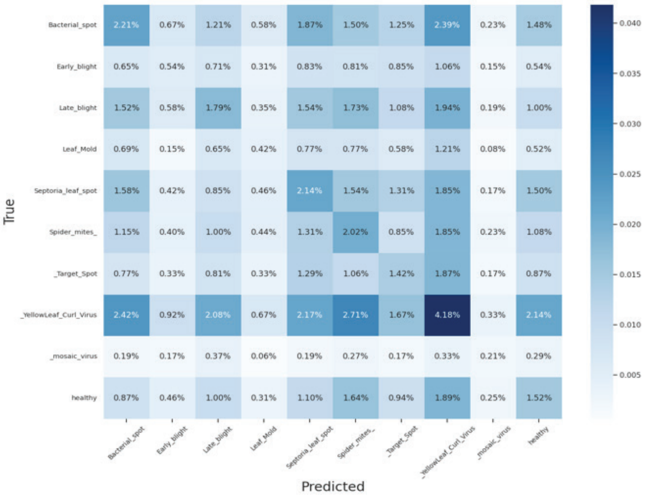

Fig. 6. Confusion matrix of the CNN model when tested on the secondary PlantVillage dataset.

We tested the same CNN model on the primary field data to check the model's generalizability. In this instance, the model's accuracy was 0.1014 and the loss was 8.1065. From a deeper evaluation using metrics like F1 Score, Precision, Recall and the confusion matrix, as in Table VI and Fig. 6, the model's performance on the primary field dataset is significantly subpar, with it failing to recognize most classes effectively. With a macro average F1 score of 4.00%, the model demonstrates a generally poor performance across all the classes. The weighted average is slightly higher, but still only at 5.00%, indicating that the model's overall performance is far from optimal for practical applications. Most of the classes like Healthy, Bacterial Spot, Early Blight, Mosaic Virus, Spider Mites, Target Spot, and Yellow Leaf Curl Virus have F1 Scores, Precision, and Recall values all at 0.00%. This suggests that the model failed entirely to identify or correctly predict these diseases.

Table VI: Evaluation Metrics When Testing the CNN On Primary Dataset

Class	F1 Score	Precision	Recall
Healthy	0.00%	0.00%	0.00%
Bacterial Spot	0.00%	0.00%	0.00%
Early Blight	0.00%	0.00%	0.00%
Late Blight	8.70%	6.67%	12.50%
Leaf Mold	6.25%	13.33%	4.08%
Mosaic Virus	0.00%	0.00%	0.00%
Septoria Leaf Spot	21.43%	12.43%	77.78%
Spider Mites	0.00%	0.00%	0.00%
Target Spot	0.00%	0.00%	0.00%
Yellow Leaf Curl Virus	0.00%	0.00%	0.00%
Macro Average	4.00%	4.00%	10.00%
Weighted Average	5.00%	5.00%	12.00%

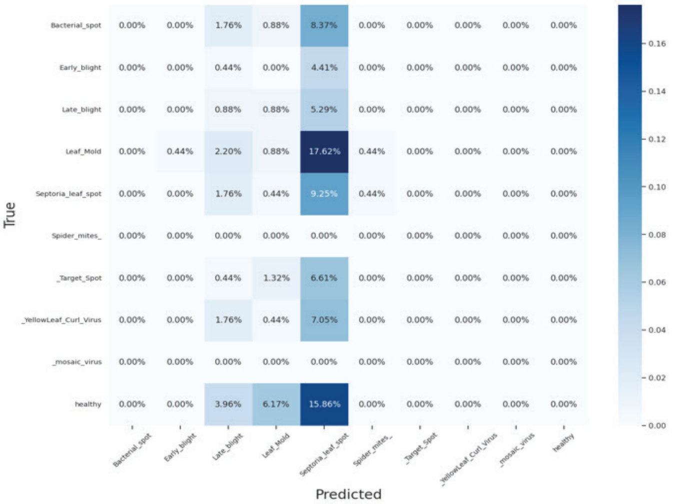

Fig. 7. Confusion matrix of the CNN model when tested on the primary dataset.

The highest F1 score is for "Septoria Leaf Spot" at 21.43%. Interestingly, while its precision is relatively low at 12.43%, its recall is notably high at 77.78%. This indicates that while the model often misclassifies other diseases as Septoria Leaf Spot (low precision), it very rarely misses an actual case of Septoria Leaf Spot (high recall). There are only two other classes, apart from Septoria Leaf Spot, where the model had non-zero scores. For "Late Blight," the model's recall (12.50%) is higher than its precision (6.67%), suggesting that it may often misclassify other conditions as Late Blight. For "Leaf Mold," the model's precision of 13.33% is higher than its recall of 4.08%, implying that when the model predicts Leaf Mold, it is more likely to be correct, but it often misses actual cases. The fact that seven out of ten classes have zero scores in all metrics indicates potential issues. The model may be struggling with the dataset, which might have imbalances, or the features of these diseases in the primary field dataset might be vastly different from the secondary PlantVillage dataset on which the model was trained.

B. MobileNetV3 Model

During training, the model ran for 16 epochs for 118 seconds and Fig. 7 and Fig. 8 show the performance of the model during training. The training accuracy was 0.9082 and the training loss was 0.3605, while the validation accuracy was 0.9158 and the validation loss was 0.3488. Since the differences between these metrics are minimal, it suggests proficient model generalization, indicating its capability as a classifier. Accuracy and loss were interpreted with caution because the dataset is heavily imbalanced. Nonetheless, the high accuracy gives a good indication of the model's performance.

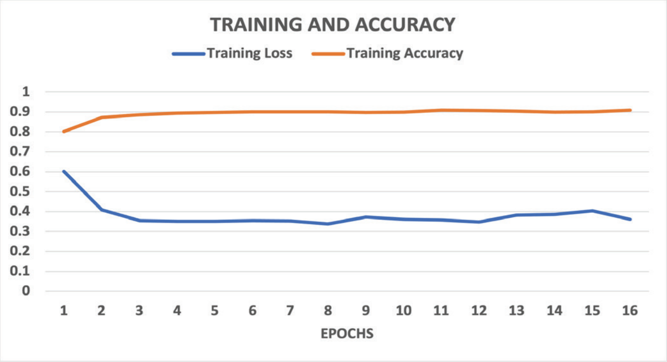

Fig. 8. Graphs of accuracy vs loss during training for MobileNetV3 model

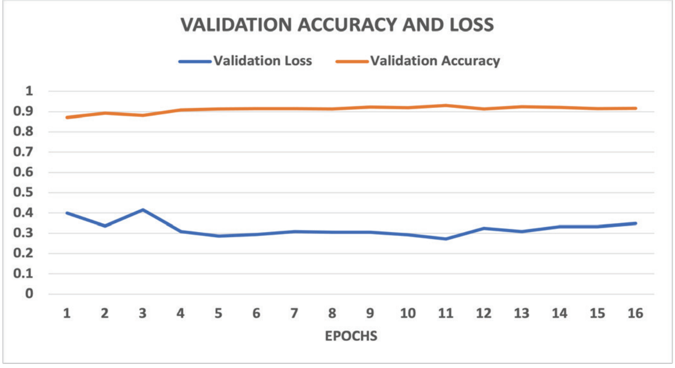

Fig. 9. Graphs of accuracy vs loss during validation for MobileNetV3 model

During testing using the secondary PlantVillage dataset, this model achieved an accuracy of 0.9259 and a loss of 0.2805. A deeper evaluation using metrics like F1 Score, Precision, and Recall and the confusion matrix (as shown in Table VII and Fig. 9) reveals that "Yellow Leaf Curl Virus" achieved the highest recorded F1 score at 22.25%, suggesting the model's strongest performance in detecting this disease. However, the fact that all scores are below 25% implies room for substantial improvement. A balanced F1 score, precision, and recall is crucial for ensuring that the model not only identifies positive cases but also reduces false identifications. The differences in these metrics for different classes may hint at dataset imbalances or inherent challenges in distinguishing certain diseases due to feature similarities.

Table VII: Evaluation Metrics When Testing the MobileNetV3 on Secondary PlantVillage Dataset

Class	F1 Score	Precision	Recall
Healthy	16.07%	16.31%	15.83%
Bacterial Spot	17.81%	18.21%	17.42%
Early Blight	10.96%	11.68%	10.32%
Late Blight	18.05%	17.99%	18.12%
Leaf Mold	11.72%	12.03%	11.43%
Mosaic Virus	10.58%	11.00%	10.19%
Septoria Leaf Spot	16.80%	16.87%	16.73%
Spider Mites	17.64%	16.98%	18.35%
Target Spot	14.30%	13.19%	15.62%
Yellow Leaf Curl Virus	22.25%	22.47%	22.03%
Macro Average	16.00%	16.00%	16.00%
Weighted Average	17.00%	17.00%	17.00%

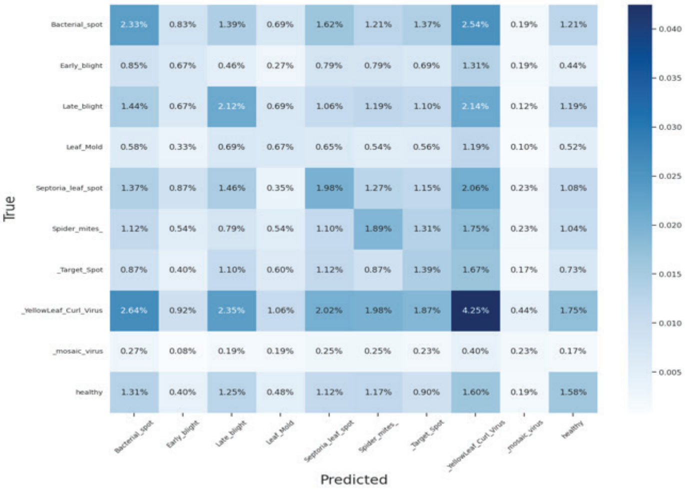

Fig. 10. Confusion matrix of the MobileNetV3 model when tested on the secondary PlantVillage dataset

We also evaluated the same CNN model on primary field data to assess its generalization capability. Here, the model's performance drastically dropped, resulting in an accuracy of only 0.1014 and a significant loss of 8.1065. A deeper evaluation using metrics like F1 Score, Precision, and Recall and the confusion matrix, as in Table VIII and Fig. 10, shows substantially lower F1 scores, precision, and recall figures, especially in classes such as "Healthy," "Bacterial Spot," and "Early Blight," underscore the model's struggles when predicting these classes on primary field data. The weighted average of 4.00% in the F1 score further reiterates the model's poor performance across classes considering their distribution.

Table VIII: Evaluation Metrics When Testing the MobileNetV3 on Primary Dataset

Class	F1 Score	Precision	Recall
Healthy	0.00%	0.00%	0.00%
Bacterial Spot	4.88%	6.25%	4.00%
Early Blight	20.00%	22.22%	18.18%
Late Blight	0.00%	0.00%	0.00%
Leaf Mold	0.00%	0.00%	0.00%
Mosaic Virus	0.00%	0.00%	0.00%
Septoria Leaf Spot	16.67%	9.93%	51.85%
Spider Mites	0.00%	0.00%	0.00%
Target Spot	0.00%	0.00%	0.00%
Yellow Leaf Curl Virus	8.16%	7.14%	9.52%
Macro Average	6.00%	6.00%	10.00%
Weighted Average	4.00%	4.00%	9.00%

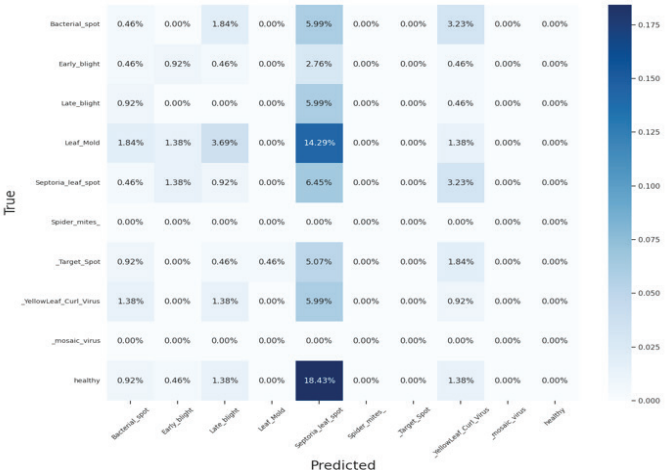

Fig. 11. Confusion matrix of the MobileNetV3 model when tested on the primary dataset

V. DISCUSSION

This chapter discusses the results of the research in relation to the objectives. The chapter also considers the performance of models trained and the comparison of model performance on secondary and primary datasets. Additionally, the chapter examines the influence of dataset imbalance, model generalization, and overfitting problems. Finally, the chapter discusses the general limitations of the study.

Table IX: A Summary of Performance Metrics (Accuracy and Loss) of the CNN and MobileNetV3 Models During Training, Validation, and Testing

Metrics	CNN		MobileNetV3	
	Accuracy (%)	Loss	Accuracy (%)	Loss
Training Performance	89.04	0.3322	90.82	0.3605
Validation Performance	83.71	0.4800	91.58	0.3488
Testing on Secondary Dataset	86.03	0.4135	92.59	0.2805
Testing on Primary Field Dataset	10.14	8.1065	9.22	12.9114

A. General Discussion

As shown in Table IX, the MobileNetV3 model exhibits a higher validation accuracy than the CNN model as MobileNetV3's validation accuracy is greater than its training accuracy. This implies that the model performs well at generalization. Additionally, the validation loss and training loss of the MobileNetV3 model converged to similar values suggesting the model is sufficiently generalizable. Furthermore, among all the models trained, this model notably achieves the highest testing accuracy and lowest testing loss. Neither model executed satisfactory performance when tested on the primary field data. They both have an accuracy of less than 0.2 and a testing or generalization greater than 1, indicating overfitting to the training data. The proposed models performed poorly at detecting tomato leaf diseases on fresh tomato leaves from the field. This finding is consistent with the research conducted by Liu & Wang [20], who encountered challenges when applying disease detection models to real-world field conditions due to variations in lighting, environmental factors, and leaf orientations.

The MobileNetV3 model achieved the highest recall value of 17%. This still confirms the assertion that the MobileNetV3 model is better at detecting tomato leaf diseases. A tomato leaf disease detection model (system) with a low recall implies that the affected leaves could be misclassified as healthy. This can lead to the spread of the disease and, ultimately, result in low crop yield. This observation aligns with the low percentages of precision and F1 score. The importance of achieving higher recall values to minimize misclassifications and improve disease detection accuracy should be emphasized.

In this study, the MobileNetV3 models have the lowest training times as well as the lowest number of epochs. These findings indicate that the MobileNetV3 architecture is computationally efficient and can achieve satisfactory results with reduced training times and iterations. This observation is aligns with the research conducted by Qian et al. [25], Vong & Chanchotisatien [26], and Wibowo et al. [27] who highlighted the efficiency and effectiveness of the MobileNetV3 model in various computer vision tasks.

The observed model performance of the models suggests potential overfitting due to limited variability. When the training data lacks sufficient diversity, models may prioritize memorizing specific instances rather than learning robust features that generalize well. While the training dataset here is large, other studies have shown that large datasets can struggle with diversity and this can hinder the models' ability to achieve high performance on unseen data [28]. Secondly, the complexity of the models themselves may have contributed to overfitting. Complex models with many parameters have a higher capacity to memorize training data, making them more prone to overfitting and leading to high variance. To mitigate the overfitting problem, regularization techniques such as dropout and weight decay can be employed during the training process. These techniques help reduce model complexity and prevent over-reliance on specific features or patterns in the training data. Additionally, increasing the size and diversity of the training dataset can also help alleviate overfitting by providing more representative samples that capture the variations present in real-world scenarios. Most papers emphasize the importance of addressing overfitting through proper regularization techniques and dataset augmentation strategies to improve the generalization performance of the models. This study employed all these techniques, but the results still turned out to be less than satisfactory.

The relatively low performance of both the CNN and MobileNetV3 models could be attributed to their limitations in capturing the intricate patterns and subtle visual cues associated with tomato leaf diseases. Plant diseases often exhibit diverse symptoms, such as discoloration, spots, and deformations, which can be challenging to differentiate from healthy plant tissue. Additionally, the variability in lighting conditions, leaf orientation, and background clutter in real-world field images further compounds the difficulty of accurate disease detection. Moreover, the dataset used for training and evaluation might not have been extensive enough to capture the full range of disease variations and environmental factors. This limited dataset diversity can restrict the models' ability to generalize and recognize different disease patterns.

Another possible reason for the results could be the dataset's imbalance. The imbalance affects the performance of the classifiers, as the accuracy of a classifier only considers how well the classifier is doing in the majority class leading to potentially misleading results. This observation is similar to the findings by Agarwal, Gupta, et al. [29], and Steininger et al. [30] who highlighted the challenges posed by imbalanced datasets in multiclassification tasks.

In conclusion, the MobileNetV3 model exhibited better performance than the CNN model, but the results were still not satisfactory for practical implementation. The models performed poorly when tested on the primary field data, highlighting the challenges of real-world deployment. Dataset imbalance also affected the model performance, and overfitting problems were observed in some cases. These findings emphasize the need for further research and improvement in data augmentation techniques and model design to develop accurate and efficient tomato leaf disease detection systems.

B. Limitations and Challenges

The data augmentation methods used in this research were mostly geometric and color transformations and these have their own limitations. Other data augmentation methods have been proven to be more effective at synthesizing new images for classification tasks. For example, feature space augmentation [16], [31]. This research originally proposed to use feature space augmentation methods like Generative Adversarial Networks (GANs) to generate new images with different leaf severities, scales, and shapes. GANs can produce fresh images of high quality that accurately reflect the characteristics of the original data [32], [33]. Datasets obtained from using GANs are usually more diverse, and the models trained on such data have really good performance.

One of the biggest gaps in literature when it comes to the classification of leaf diseases is that the datasets used usually have images that are only captured at one physiological state of the leaf. The images found in the PlantVillage dataset are only of mature leaves and there are no old or young leaves available, which means the dataset that is used is limited. Unfortunately, training of GANs is computationally expensive and unstable. It takes a lot of time and resources to fine-tune hyper-parameters and balance the dynamics between the discriminator and generator of a GAN. Given the constraints of computational resources common in academic research, simpler data augmentation methods were employed to obtain the data used in this study. From the results, it is noted that this did not improve the performance of the models significantly.

A further challenge in this research arose during the testing of the trained models with primary field data. The PlantVillage dataset was captured under controlled conditions and the primary field dataset that was used to test the models had different capture conditions. The fresh images were collected by different devices as well as individuals. They were also captured in different light conditions and at different angles. Despite the use of data augmentation to simulate such real-world conditions, the trained models exhibited poor generalization and struggled to accurately predict disease on the fresh images. This challenge persists in several studies [34]. However, this research contributes towards overcoming this hurdle by providing a valuable dataset that can enhance future research in this field.

VI. CONCLUSION AND FUTURE DIRECTIONS

A. Conclusion

The pre-trained model, MobileNetV3, performed better than the CNN model at detecting tomato leaf diseases in primary field data. The MobileNetV3 model was faster at training and had better accuracy and a better recall than the CNN model. Despite the MobileNetV3 performing better than the CNN model, its metrics did not give confidence to the generalization ability of the model in the task for tomato leaf disease classification. Additionally, with an accuracy of 92.59% and a loss of 0.2805, the pre-trained MobileNetV3 model that was trained on augmented data conventionally performed better than the CNN model. In detecting tomato leaf diseases in primary field data, it achieved an accuracy of 9.2% and a loss of 12. 91 whereas the CNN achieved an accuracy of 10.14% and a loss of 8.11. Regardless of whether the performance of both models did not meet the expectations of the research (that they would be good at generalization), the experiments aided in showing that the models that are trained on the PlantVillage dataset are not as effective when they are used in real-world scenarios.

A recurring limitation in research on the use of CNNs and pre-trained models and their applications in tomato leaf disease detection is a lack of use or testing of the model in other datasets. In most research, the models are trained and tested on the public PlantVillage dataset. The models always seem to perform well but their application is not well explored because of lack of data. This research, therefore, tested such models (basic CNN and pre-trained MobileNetV3 models) on fresh images of tomato leaves collected in Malawian fields. The images were few and a human consultant was engaged in diagnosing the leaves collected. Testing the proposed models on this data proved the suspicion that the models, although performing quite well during training, are not good predictors when used in real-world scenarios. There is no argument that advanced technological advances are needed to help improve productivity in Malawi's agricultural sector, but more still needs to be explored in order to have a system that can be used in Malawian farms.

B. Future Directions

Malawi aims to enhance agricultural production and innovation as part of its post-COVID-19 socio-economic recovery and to achieve its vision of self-reliance and prosperity as a nation [35], [36]. Recognizing that an innovative and productive agricultural sector is key to industrial growth, the government views technological integration into agricultural practices as a crucial strategy. This focus is particularly relevant given the scarcity of expert human diagnostics in Malawi which necessitates the development of computational tools for disease diagnosis in crops [23].

This study lays the groundwork for creating a mobile-based diagnostic system for identifying tomato leaf diseases through image analysis. Although the initial outcomes were not fully satisfactory, they provide a foundation for future research aimed at refining these diagnostic tools for practical use. Research priorities could be:

- Investigating advanced data augmentation techniques to address the current lack of dataset diversity, improving model performance despite higher computational demands.
- Exploring how different camera resolutions affect diagnostic accuracy and the optimization of image capture for reliable disease identification.
- Creating systems that recommend treatment options based on diagnosed diseases using natural language processing.
- Extending the diagnostic system to additional crops, which could significantly aid Malawi's farmers, aligning with national goals for agricultural efficiency and sustainability.

References

[1] P. N. Eviness, M. Charles, and K. Hilda, "An assessment of tomato production practices among rural farmers in major tomato growing districts in Malawi," *Afr J Agric Res*, vol. 18, no. 3, pp. 194–206, Mar. 2022, doi: 10.5897/ajar2021.15893.

[2] V. L. Flax, C. Thakwalakwa, C. H. Schnefke, J. C. Phuka, and L. M. Jaacks, "Food purchasing decisions of Malawian mothers with young children in households experiencing the nutrition transition," *Appetite*, vol. 156, no. August 2020, p. 104855, 2021, doi: 10.1016/j.appet.2020.104855.

[3] I. R. Fandika, D. Kadyampakeni, and S. Zingore, "Performance of bucket drip irrigation powered by treadle pump on tomato and maize/bean production in Malawi," *Irrig Sci*, vol. 30, no. 1, pp. 57–68, 2012, doi: 10.1007/s00271-010-0260-2.

[4] N. Mango, L. Mapemba, H. Tchale, C. Makate, N. Dunjana, and M. Lundy, "Comparative analysis of tomato value chain competitiveness in selected areas of Malawi and Mozambique," *Cogent Economics and Finance*, vol. 3, no. 1, Sep. 2015, doi: 10.1080/23322039.2015.1088429.

[5] B. N. Nyoni *et al.*, "Pyrethroid resistance in the tomato red spider mite, Tetranychus evansi, is associated with mutation of the para-type sodium channel," no. March, pp. 891–897, 2011, doi: 10.1002/ps.2145.

[6] X. Wang, J. Liu, and G. Liu, "Diseases Detection of Occlusion and Overlapping Tomato Leaves Based on Deep Learning," *Front Plant Sci*, vol. 12, Dec. 2021, doi: 10.3389/fpls.2021.792244.

[7] C. Smart and R. M. Davis, "CHAPTER 14: Diagnostic Methods for Identifying Tomato Diseases," in *Tomato Health Management*, The American Phytopathological Society, 2017, pp. 135–144. doi: 10.1094/9780890544884.014.

[8] Y. Yang *et al.*, "A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions," *Comput Biol Med*, vol. 139, Dec. 2021, doi: 10.1016/j.compbiomed.2021.104887.

[9] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture (Switzerland)*, vol. 11, no. 8. MDPI AG, Aug. 01, 2021. doi: 10.3390/agriculture11080707.

[10] H. Mayfield, C. Smith, M. Gallagher, and M. Hockings, "Use of freely available datasets and machine learning methods in predicting deforestation," *Environmental Modelling & Software*, vol. 87, pp. 17–28, Jan. 2017, doi: 10.1016/j.envsoft.2016.10.006.

[11] B. Farnham, S. Tokyo, B. Boston, F. Sebastopol, and T. Beijing, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems," 2019.

[12] V. Silaparasetty, *Deep Learning Projects Using TensorFlow 2*. Apress, 2020. doi: 10.1007/978-1-4842-5802-6.

[13] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[14] A. A. Awan, "A Complete Guide to Data Augmentation," *DataCamp*, 2022. Available: https://www.datacamp.com/tutorial/complete-guide-data-augmentation (accessed Jun. 10, 2023).

[15] N. K. E, K. M, P. P, A. R, and V. S, "Tomato Leaf Disease Detection using Convolutional Neural Network with Data Augmentation," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jun. 2020, pp. 1125–1132. doi: 10.1109/ICCES48766.2020.9138030.

[16] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[17] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, New York, NY, USA: ACM, Jan. 2020, pp. 249–253. doi: 10.1145/3371158.3371196.

[18] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 293–301. doi: 10.1016/j.procs.2020.03.225.

[19] S. Nandhini and K. Ashokkumar, "Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network," *Multimed Tools Appl*, vol. 80, no. 12, pp. 18583–18610, May 2021, doi: 10.1007/s11042-021-10599-4.

[20] J. Liu and X. Wang, "Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model," *Plant Methods*, vol. 16, no. 1, Jun. 2020, doi: 10.1186/s13007-020-00624-2.

[21] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, and Y. Hu, "Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet," *Comput Electron Agric*, vol. 178, Nov. 2020, doi: 10.1016/j.compag.2020.105730.

[22] V. Gonzalez-huitron, A. Le, L. E. Amabilis-sosa, B. Ramírez-pereda, and H. Rodriguez, "Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4," *Comput Electron Agric*, vol. 181, no. January, 2021, doi: 10.1016/j.compag.2020.105951.

[23] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics." [Online]. Available: http://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf!!!

[24] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[25] S. Qian, C. Ning, and Y. Hu, "MobileNetV3 for Image Classification," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, IEEE, Mar. 2021, pp. 490–497. doi: 10.1109/ICBAIE52039.2021.9389905.

[26] C. Vong and P. Chanchotisatien, "Mobile-based Application for COVID-19 Detection from Lung X-Ray Scans with Artificial Neural Networks (ANN)," in *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, IEEE, Jan. 2022, pp. 232–237. doi: 10.1109/ECTIDAMTNCON53731.2022.9720351.

[27] A. Wibowo, S. R. Purnama, P. W. Wirawan, and H. Rasyidi, "Lightweight encoder-decoder model for automatic skin lesion segmentation," *Inform Med Unlocked*, vol. 25, p. 100640, 2021, doi: 10.1016/j.imu.2021.100640.

[28] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.

[29] M. Agarwal, S. K. Gupta, and K. K. Biswas, "Development of Efficient CNN model for Tomato crop disease identification," *Sustainable Computing: Informatics and Systems*, vol. 28, Dec. 2020, doi: 10.1016/j.suscom.2020.100407.

[30] M. Steininger, K. Kobs, P. Davidson, A. Krause, and A. Hotho, "Density-based weighting for imbalanced regression," *Mach Learn*, vol. 110, no. 8, pp. 2187–2211, Aug. 2021, doi: 10.1007/s10994-021-06023-5.

[31] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput Electron Agric*, vol. 187, Aug. 2021, doi: 10.1016/j.compag.2021.106279.

[32] Y. Kurmi and S. Gangwar, "A leaf image localization based algorithm for different crops disease classification," *Information Processing in Agriculture*, no. xxxx, pp. 1–19, 2021, doi: 10.1016/j.inpa.2021.03.001.

[33] F. Wang *et al.*, "Generative adversarial networks and convolutional neural networks based weather classification model for day ahead short-term photovoltaic power forecasting," *Energy Convers Manag*, vol. 181, pp. 443–462, Feb. 2019, doi: 10.1016/j.enconman.2018.11.074.

[34] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst Eng*, vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[35] G. of Malawi, "Malawi COVID-19 Socio-Economic Recovery Plan: 2021-2023," Ministry of Economic Planning and Development and Public Sector Reforms, Lilongwe, Malawi. MALAWI, 2021.

[36] National Planning Commission, "Malawi's Vision, An Inclusively Wealthy and Self-reliant Nation: Malawi 2063," pp. 1–92, 2020.

Lindizgani Kingstone Ndovie: Graduate in Master of Science in Data Science, specializing in biostatistics, from the African Centre of Excellence in Data Science (ACE-DS), University of Rwanda. She also holds a Bachelor of Science, specializing in statistics, from the University of Malawi. With over five years of experience in research, monitoring and evaluation, and data management, she has contributed significantly to the public health and biomedical research sectors in Malawi. Her research interests include GIS, disease detection, and the application of data science methodologies to inform strategic decision-making in public health.

Emmanuel Masabo: Senior Lecturer at the University of Rwanda, School of ICT and the Head of Research in the African Centre of Excellence in Data Science (ACEDS) - University of Rwanda (UR). He holds a PhD in Software Engineering from Makerere University (Uganda), a Master of Engineering in Computer Application Technology degree from Central South University, China, and a Bachelor of Science in Computer Engineering and Information Technology from what was formerly the Kigali Institute of Science and Technology (currently the College of Science and Technology, University of Rwanda). He has working experience in academia and public institutions and has published various papers in leading international journals and conferences. His main research interests are Artificial Intelligence (AI), Machine Learning, Cyber-security, Big Data, Software Engineering, and the Internet of Things (IoT).

CONTENTS PAGE