Tanzania Journal of Science 48(4): 913-926, 2022
ISSN 0856-1761, e-ISSN 2507-7961

Application of MobileNets Convolutional Neural Network Model in Detecting Tomato Late Blight Disease

Richard C Rajabu[1*], Juma S Ally[2] and Jamal F Banzi[3]
[1]Mbeya University of Science and Technology, College of Information Communication Technology, Department of Computer Science and Engineering, P.O. Box 131, Mbeya, Tanzania.
[2]Mbeya University of Science and Technology, College of Information Communication Technology, Department of Electronics and Telecommunication, P.O. Box 131, Mbeya, Tanzania.
[3]Sokoine University of Agriculture, Department of Tourism and Recreation, P.O. Box 3167, Morogoro, Tanzania.
Emails: RCRajabu@gmail.com, jeiside@gmail.com, jamalbanzi@sua.ac.tz
[*]Corresponding author
Received 12 Oct 2022, Revised 21 Dec 2022, Accepted 22 Dec 2022, Published Dec 2022
DOI: https://dx.doi.org/10.4314/tjs.v48i4.17

Abstract

Late blight (LB) disease causes significant annual losses in tomato production. Early identification of this disease is crucial in halting its severity. This study aimed to leverage the strength of Convolutional Neural Networks (CNNs) in automated prediction of tomato LB. Through transfer learning, the MobileNetV3 model was trained on high-quality, well-labeled images from Kaggle datasets. The trained model was tested on different images of healthy and infected leaves taken from different real-world locations in Mbeya, Arusha, and Morogoro. Test results demonstrated the model's success in identifying LB disease, with an accuracy of 81% and a precision of 76%. The trained model has the potential to be integrated into an offline mobile app for real-time use, improving the efficiency and effectiveness of LB disease detection in tomato production. Similar methods could also be applied to detect other tomato infections.

Keywords: MobileNets; convolutional neural networks; plant diseases detection; image classification; transfer learning.

Introduction

Tomato (*Solanum lycopersicum*) is one of the dominant vegetable crops produced worldwide (Costa and Heuvelink 2018). In Tanzania, it is among the highly produced fruits and vegetables and mostly cultivated by smallholder farmers (NBS 2017, NBS 2021). It is a priority to these farmers because they can potentially generate high profits per area due to multiple harvests (Mutayoba and Ngaruko 2018). Production of tomatoes is affected by the prevalence of pests and diseases including the late blight (LB) disease (Mutayoba and Ngaruko 2018). This disease causes significant yield losses annually (Meya et al. 2014, Nowicki et al. 2013). Traditional methods of detecting this disease involve visual assessments and laboratory techniques such as Polymerase Chain Reaction (PCR), Enzyme Linked Immune Sorbent Assay (ELISA), fluorescence in situ hybridization and biomarker-based detection technology (Sankaran et al. 2010). These methods are complex, time consuming and most of time not precise (Xie et al. 2015, Francis and Deisy 2019). Furthermore, inadequate extension services, poor and fewer facilities such as laboratories and poor communication make it difficult for farmers to obtain prompt services (Benard et al. 2014). Therefore, the availability of a cheap and reliable technology for the timely detection of LB disease will help in its effective management.

Convolutional Neural Networks (CNN) are highly capable of extracting features from data through convolution structures, they have achieved a number of groundbreaking results in Computer Vision and Natural Language Processing (Li et al. 2021). The ubiquity of mobile devices and embedded systems has led to the development of efficient CNN models optimized for mobile devices (Howard et al. 2019). These models are lightweight, they require minimal computing resources to operate. There is a real potential for these models to be deployed in several real-time scenarios including crop disease diagnosis. MobileNet models are a class of lightweight CNN models optimized for mobile devices and embedded systems (Howard et al. 2017).

Several studies have applied the MobileNet model in detecting plant leaf diseases. These include Kulkarni (2018), Chen et al. (2021) and Bi et al. (2022) which compared the performance of MobileNetV1 and other standard CNN models including InceptionV3, ResNet152, and VGG16 in leaf disease detection. The MobileNet model achieved good accuracy values despite being a smaller model compared to the other models. Others including Huang et al. (2020), Rajbongshi et al. (2020), Elfatimi et al. (2022), Zaki et al. (2020) and Mukhtar et al. (2021) applied this model in leaf disease detection and attained test accuracy values as high as 96%. Moreover, a number of studies have tried to improve this model's performance in plant disease detection by integrating it with other machine learning models and techniques. Ashwinkumar et al. (2022) integrated it with a bilateral filter for image quality enhancement and Extreme Learning Machine (ELM) for disease classification, Nguyen and Paik (2020) combined it with traditional machine learning algorithms, Multi-layer Perceptron (MLP), Support Vector Machines (SVMs) and Random Forests (RFs) in order to improve feature classification. Dan et al. (2019) embedded Squeeze-and-Exception Networks (SENet) module between the last pooling layer and convolution layer of MobileNetV2 network. These studies reported test accuracies of above 95%. However, most of these studies did not address the generalizability of this model in real life situations, and most of them applied images taken from controlled environments, hence the results obtained can be misleading. On the other hand, there are some studies such as Elfatimi et al. (2022) that applied real world images for training and testing but the images used are of very similar nature and background and hence it is not possible to tell if the model will generalize in dissimilar environments.

This paper elucidates development of lightweight CNN models to perform tomato diseases detection using simple leaf images of tomato. It presents assistive technology to farmers to compliment extension officers in dealing with the tomato LB disease. By leveraging MobileNetV3, the model was trained to identify features of a disease on high quality tomato leaf images. The disease is characterized by lesions on the leaves that start as undefined water-soaked dots and quickly become pale green to brownish-black in color (Nelson 2008). To see how the model performs in different situations, it was tested on real world images with complex backgrounds and varying levels of quality.

The main contributions of this study are: (1) a versatile model for detecting LB disease on tomato leaves, which can be integrated into a mobile app for offline use, and (2) a dataset of tomato leaf images for use in future studies.

Materials and Methods

This section presents the methodology used to train and analyze the performance of the MobileNetV3 model in detecting tomato plant LB disease.

Experimental setup

Experiments were run on a laptop with an Intel Core i7 processor and no GPU. Python language was used for programming, TensorFlow and Keras libraries for implementing and training the neural networks. Test images were taken with various smartphones.

Data used

This study used public data from Kaggle Datasets. About 5300 images of healthy and LB infected were gathered and used for training and validation, these images are of a high quality with a resolution of 96 pixels per inch, taken from a controlled environment characterized by a simple (uniform) background (Figure 1). They can be found at (https://www.kaggle.com/datasets/emmarex/plantdisease) and (https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf).

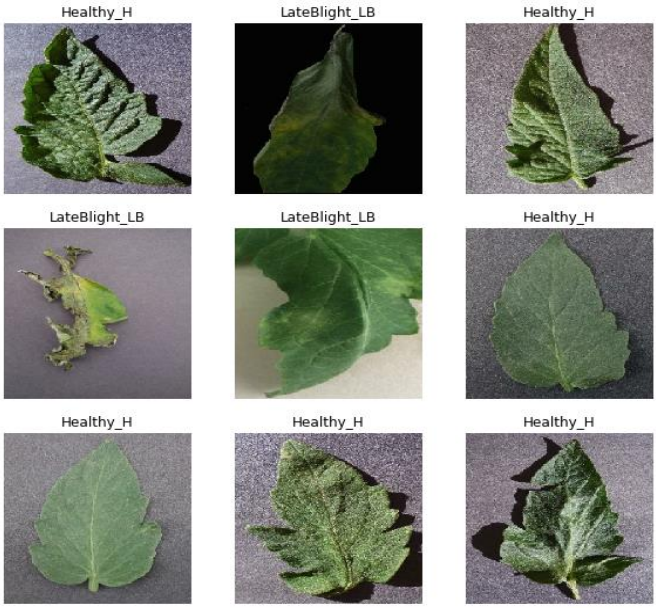

Figure 1: Samples of labelled images used for training and validation.

The test data contained the images taken from Mbeya, Arusha, Morogoro and the web. The study was able to collect about 1478 images from different field conditions. These images have complex backgrounds with varying qualities and resolutions, mostly acquired by cameras of different mobile phones. They test the versatility of the trained model and assist in fine-tuning it based on real-life features present in these images. Figure 2 shows a sample of test images randomly obtained to represent various real-world scenarios, including shadows, poor lighting, different illumination intensities, and complex (non-uniform) backgrounds.

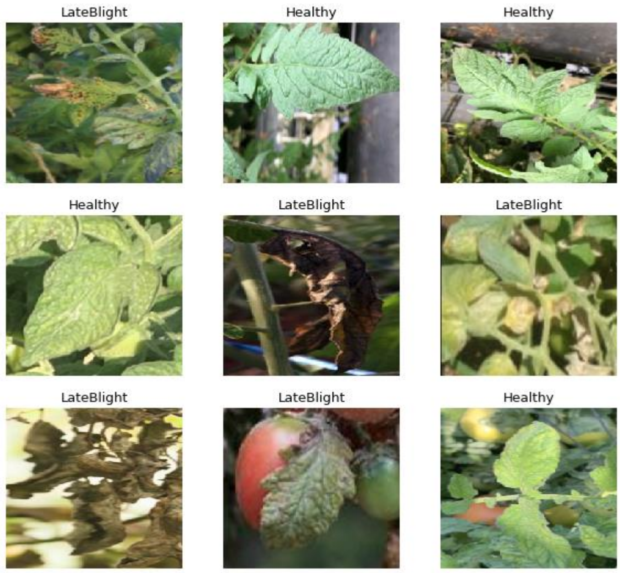

Figure 2: Samples of labelled test images gathered from farms and the web.

The MobileNetV3 model

MobileNetV3 is the latest version of the MobileNet models which are a class of lightweight CNN models optimized for mobile devices and embedded systems (Howard et al. 2017). MobileNetV3 is built upon Depthwise separable convolutions introduced in MobileNetV1, Linear bottleneck and inverted residual structure introduced in MobileNetV2 and Lightweight attention modules (LAM) based on squeeze and excitation into the bottleneck structure. This concept was introduced by Mobile neural architecture search Networks (Howard et al. 2019). LAM focuses the model's attention on the most useful parts of an image. These features together with the use of the ReLu6: $\frac{\text{ReLu6}(x+3)}{6}$ activation function make this model highly efficient in terms of computational requirements and model accuracy (Howard et al. 2019). Therefore, this model can be easily integrated in mobile apps and work efficiently in mobile devices with limited computational resources such as low memory and power (Qian et al. 2021). MobileNetV3 achieves State of The Art results for lightweight models in key computer vision problems (Howard et al. 2019).

Transfer Learning

This study leverages the MobileNetV3 model in classifying healthy and diseased tomato leaves. In doing so, the concept of transfer learning (TL) is applied, whereby parameters from an existing (pre-trained) model are modified to ally with a specific problem. Pre-trained model is preferred over training a new model since training process requires powerful resources, a large number of well labelled data and it is time consuming. Originally MobileNetV3 was trained to classify 1000 classes of objects on the image database, ImageNet. In this study, it is retrained to perform a binary classification of healthy and diseased tomato leaves. Figure 3 details the TL workflow adopted in the experiments. The model is trained by modifying (fine-tuning) the hyper-parameter (Table 1) and its trainable parameters until a sufficient test accuracy is attained. Hyper-parameters are the parameters which are not part of the model's architecture, model's parameters are part of the model's architecture, these include the number of layers, weights, and biases. Trainable parameters can change their values during training while non-trainable parameters cannot.

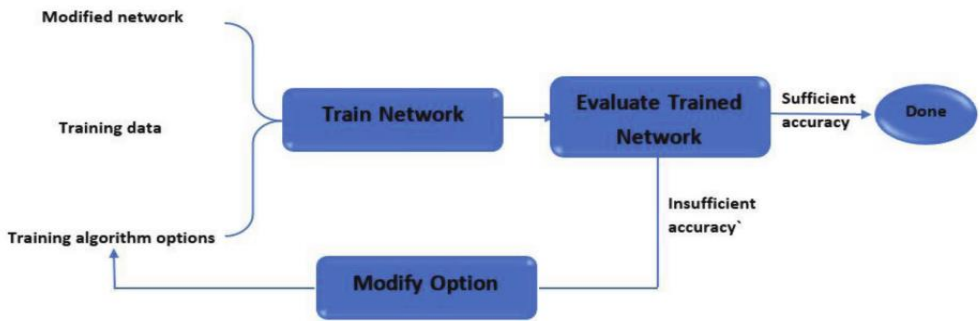

Figure 3: Adopted workflow following the TL method of model training and modification.

MobileNetV3 has two models: MobileNetV3-Large and MobileNetV3-Small. This study applies the MobileNetV3-Small model which has the fewest parameters and consequently more efficient for mobile devices. The model is first modified to perform a binary classification task and then trained on the data using the set hyper-parameters. Through TL the model is modified to perform this binary classification task. Figure 4 represents the architecture of the modified model whereby MobileNetV3small is used as the feature extractor (FE). The role of the FE is to recognize patterns pertaining to healthy and diseased leaves from an image.

In Figure 4, the input layer receives coloured images with shape 224 x 224 x 3 (224 pixels x 224 pixels x 3 colour channels) all images were resized to this size for the model to process them. Resizing was done using the Python Imaging Library (PIL). In this study, the resized images are first passed to the augmentation layer, data augmentation is the technique used to slightly alter the input images (such as flipping the image and rotating it) without distorting their meaning. This technique gives the model more examples to train from. Then the images are passed to the feature extractor, the MobileNetV3-Small, this learns the features from a 224 x 224 x 3 image and produces a 7 x 7 x 576 feature map, the pooling layer summarizes these features, it reduces the feature map to 576 values which are used by the prediction layer to predict the presence of the LB disease. The dropout layer is used to ensure that the model does not overfit the training data, it achieves this by adding noise to the input values thus making the network more robust (Srivastava et al. 2014). In the experiments the drop rate of 0.5 is used, the model generalizes better with this value compared to other tested values.

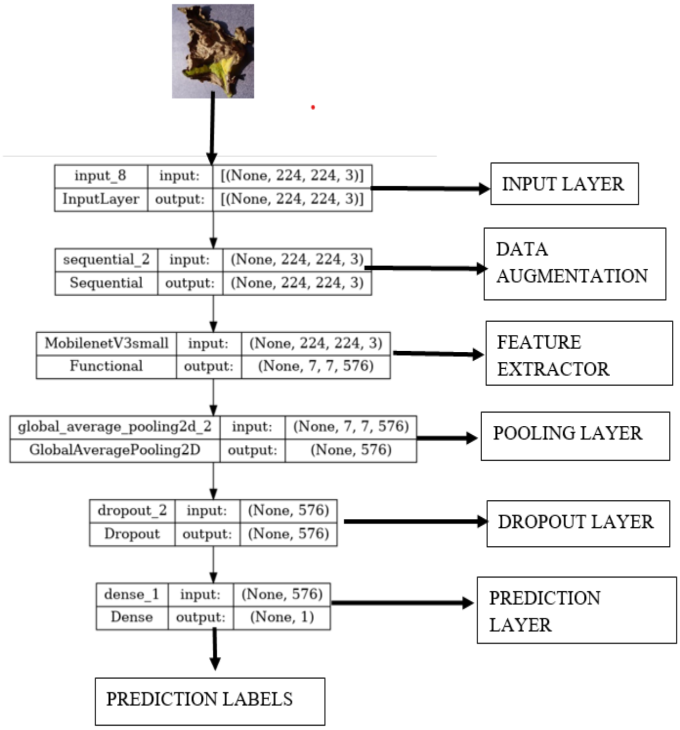

Figure 4: Model used for feature extraction and prediction.

The Adam optimizer is used in all experiments; this optimizer is computationally efficient, requires little memory (Kingma and Ba 2014) and, has achieved good results when used with the MobileNet model such as in Barman et al. (2020), Zaki et al. (2020), and Elfatimi et al. (2022). The learning rate was set at 0.000001 after higher rates resulted in significant overfitting. The model performed poorly on test data with higher learning rates. After experimenting with different numbers of epochs, 100 produced the best results for the model. This was based on its performance on the validation dataset and the use of early stopping to prevent overfitting. The choice of 100 epochs was key to the model's ability to generalize and perform well on unseen data. Binary accuracy metric was used to evaluate the performance of the model during training, it calculates how often predictions match binary labels.

Table 1: Summary of training hyper-parameters used for model training

Loss function	Optimizer	Learning rate	Metrics	Batch	Iterations
Categorical cross entropy	Adam	0.000001	Binary Accuracy	8,16 and 32	100

Following the TL work flow, the FE is frozen and then trained on the training dataset. Freezing means that the model is not able to change its existing parameters (weights and biases) during training. Freezing the FE meant that only 577 were trainable, i.e., the neurons in the prediction layer as shown in Figure 5. Based on these parameters, the model was trained on the batch sizes of 8, 16 and 32, respectively, and these values allowed the network to train efficiently.

```
Layer (type)                Output Shape              Param #
=================================================================
input_8 (InputLayer)        [(None, 224, 224, 3)]     0

sequential_2 (Sequential)   (None, 224, 224, 3)       0

MobilenetV3small (Functiona  (None, 7, 7, 576)        939120
l)

global_average_pooling2d_2   (None, 576)              0
(GlobalAveragePooling2D)

dropout_2 (Dropout)         (None, 576)               0

dense_1 (Dense)             (None, 1)                 577

=================================================================
Total params: 939,697
Trainable params: 577
Non-trainable params: 939,120
```

Figure 5: Summary of the initial model, all parameters from the MobileNetV3small model are frozen.

Batch size is the number of training examples used per each iteration. In this study, only this hyper-parameter was altered during the experiments because of its importance in determining the accuracy of a classifier (Kandel and Castelli 2020). After obtaining unsatisfactory test results, the initial model was fine-tuned. This is by unfreezing the final convolutional block (layers) of the feature extractor (Figure 6).

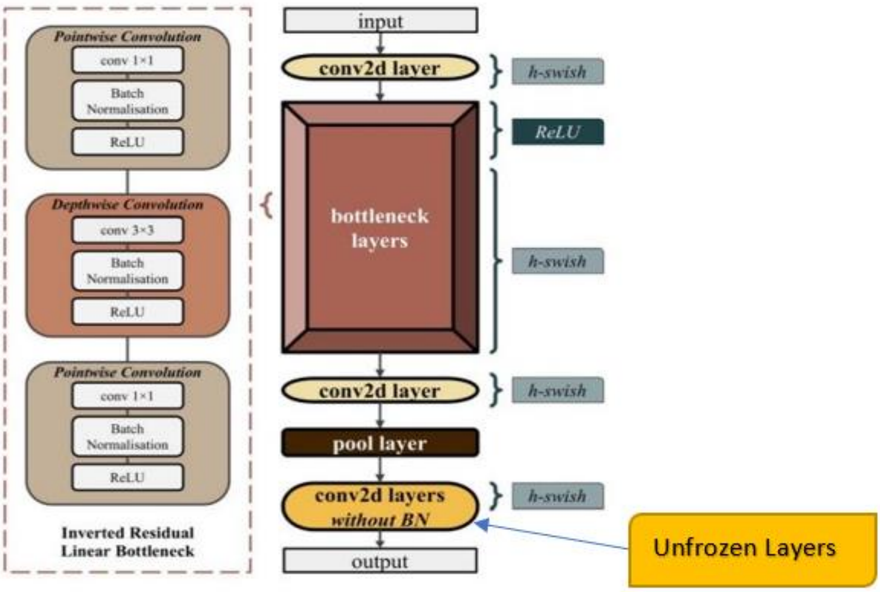

Figure 6: MobileNetV3 architecture (Qian et al. 2021) and the position of unfrozen layers.

Parameters in this layer are more capable of learning complex features in a given image. Unfreezing provides more parameters for learning key features of the disease. Figure 7 shows the number of parameters after unfreezing the FE.

```
 Layer (type)                Output Shape              Param #
=================================================================
 input_17 (InputLayer)       [(None, 224, 224, 3)]     0

 sequential_3 (Sequential)   (None, 224, 224, 3)       0

 MobilenetV3small (Functiona  (None, 7, 7, 576)        939120
 l)

 global_average_pooling2d_4   (None, 576)              0
 (GlobalAveragePooling2D)

 dropout_6 (Dropout)         (None, 576)               0

 dense_3 (Dense)             (None, 1)                 577

=================================================================
Total params: 939,697
Trainable params: 57,025
Non-trainable params: 882,672
```

Figure 7: Summary of the structure of the fine-tuned model with added trainable parameters.

The fine-tuned model is then trained on the dataset using the parameters shown in Table 1. Fine-tuned model showed significant improvements over the initial model. Further fine-tuning resulted into overfitting due to the size of the training dataset. Therefore, the fine-tuned model was selected as the best model for detecting tomato LB disease.

Results

This section presents the results of the model's performance, including validation and test accuracy, training curves, F-1 score, precision, recall, classification accuracy, and a confusion matrix. These values demonstrate how the two models performed in the experiments. Training curves show the model's performance during training and how accuracy varies with respect to loss. The results for the conducted experiments are presented in the following sections.

Initial model results

The initial model had the best test accuracy with a batch size of 16, but the training curves (Figure 8) showed a need for fine tuning the FE due to high validation loss despite high validation accuracy.

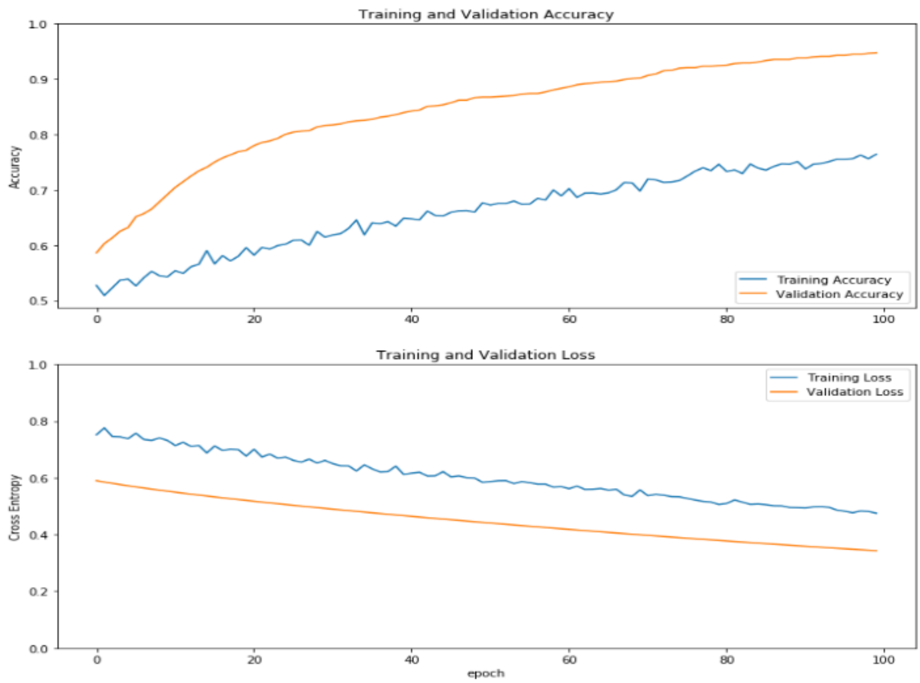

Figure 8: Training curves for the initial model.

Table 2 shows the results obtained in the three training phases, further evaluation on the initial model showed that it poorly predicted the diseased and healthy leaves, providing further support for fine-tuning the FE.

Table 2: Accuracy values obtained after training and testing the initial model

Batch size	Training accuracy	Validation accuracy	Test accuracy
8	0.7837	0.9189	0.7138
16	**0.6815**	**0.8322**	**0.7740**
32	0.6600	0.8695	0.6549

Fine-tuned model results

After fine-tuning the initial model, it was trained on the dataset with the best results using a batch size of 8. Figure 9 shows the training process. Both training and validation losses were minimized, and training and validation accuracies converged at high values, indicating minimal overfitting.

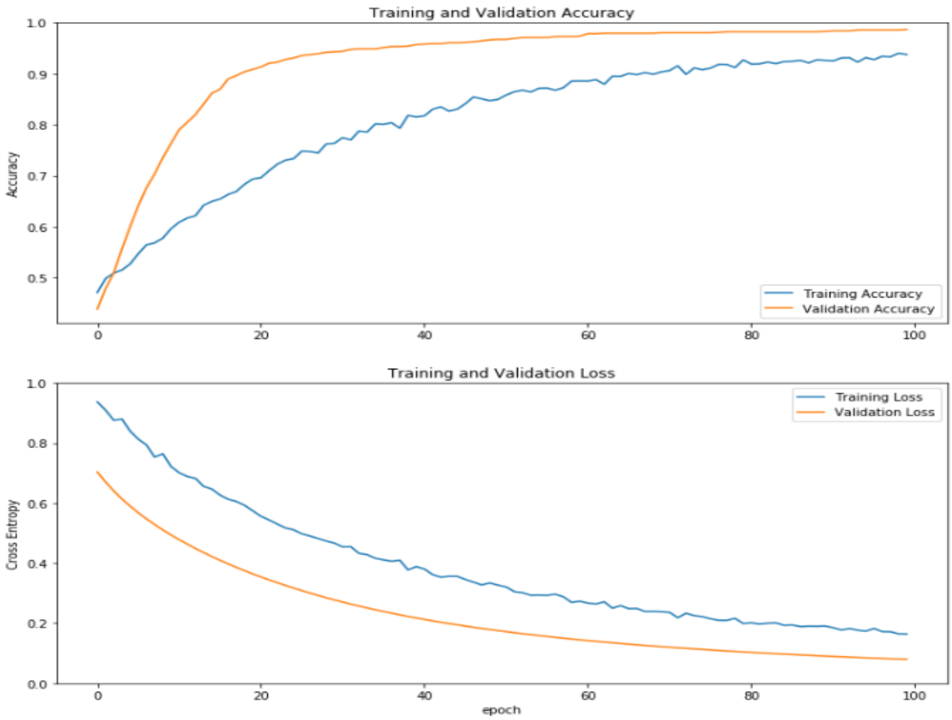

Figure 9: Training curves for the fine-tuned model.

Table 3 shows the accuracies achieved after the three experiments of training the fine-tuned model.

Table 3: Accuracy values after training and testing the fine-tuned model

Batch size	Training accuracy	Validation accuracy	Test accuracy
8	**0.9623**	**0.9842**	**0.8525**
16	0.9423	0.9832	0.8349
32	0.8580	0.9599	0.7909

Confusion matrix and classification report show a clear picture of the performance of this model in identifying diseased and healthy leaves. The confusion matrix (Figure 10) summarizes the predications made by the model. The model correctly identified 525 diseased leaves out of 554, and only misclassified 29 leaves. Furthermore, it correctly identified 300 healthy leaves out of 470, while 170 healthy leaves were misclassified as diseased.

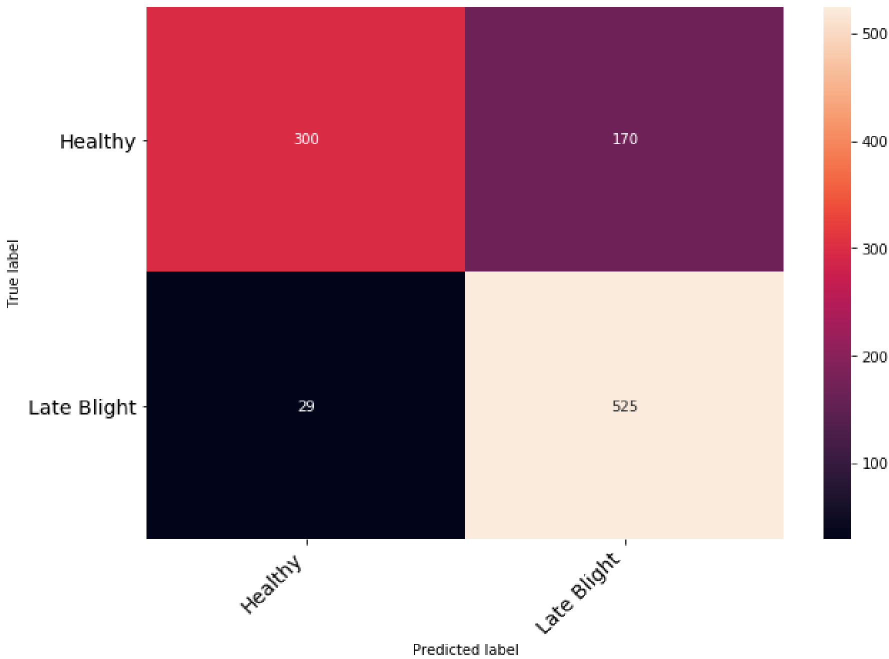

Figure 10: Confusion matrix showing the predictions made by the fine-tuned model.

A classification report (Figure 11) is used as a measure of the quality of a classification model. It clarifies the predictions made by the model by presenting them in terms of the precision, recall, F1-score and classification accuracy. These values tell how the model performs in the experiments. There were two classes of tested images, 0 for healthy leaves and 1 of diseased leaves.

	precision	recall	f1-score	support
0	0.91	0.64	0.75	470
1	0.76	0.95	0.84	554
accuracy			0.81	1024
macro avg	0.83	0.79	0.80	1024
weighted avg	0.83	0.81	0.80	1024

Figure 11: Classification report.

Precision represents the percent of positive predictions for each class, the ability of a model not to label an instance positive while it is actually not. Recall also referred to as Sensitivity or True Positive Rate (TPR) represents the ability of a model to capture all positive instances, a fraction of positives that were correctly identified. F1-score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. In general, a model with high precision and high recall will have a high F1-score, indicating good overall performance. Accuracy (Classification accuracy) represents the number of correct predictions made divided by the total number of predictions made. The model's accuracy in assigning predicted labels to test images is demonstrated in Figure 12. These labels can be used by the user to make informed decisions based on the model's feedback.

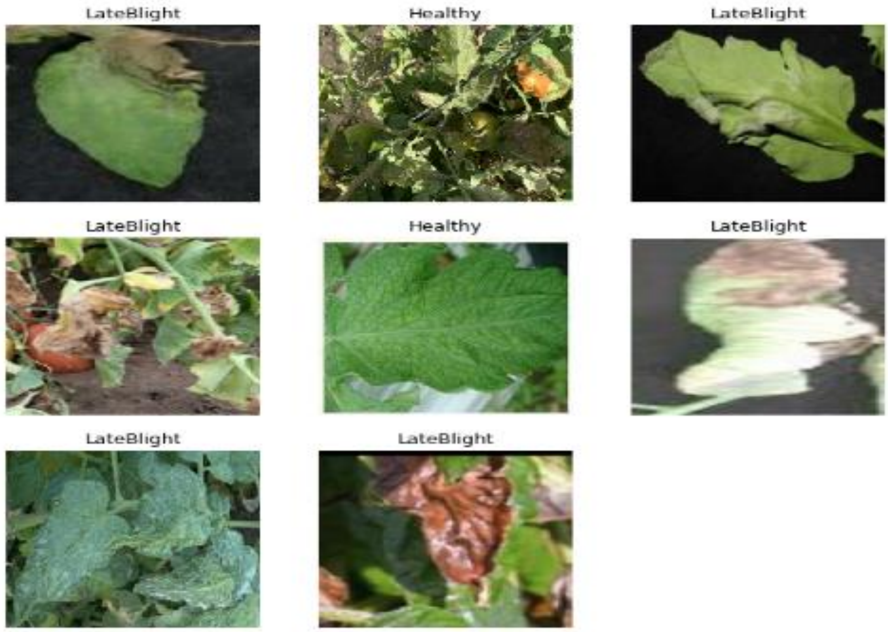

Figure 12: Samples of the prediction labels assigned by the model on the test images.

Discussions

The MobileNetV3 model demonstrated strong performance in detecting tomato late blight disease in real-world images, even in the presence of challenges such as shadows and complex backgrounds. This is a significant improvement compared to previous approaches that have struggled to accurately detect the disease in such conditions (Table 4). However, the model's performance in detecting healthy leaves from real-world images could be further improved to reduce false alarms. One potential strategy for addressing this issue could be to increase the number and diversity of healthy leaves in the training dataset. In comparison to other models that have been used for detecting late blight disease, the MobileNetV3 model offers several advantages, such as its lightweight design and efficient feature extraction. The results of this study demonstrate the potential for the MobileNetV3 model to be a powerful tool for automating the detection of tomato late blight disease. Further research could focus on optimizing the model using techniques such as LAM and increasing the diversity of the training dataset in order to improve its precision and accuracy.

Table 4: Performance of the proposed method compared to other studies in detecting tomato LB disease

S/No	Study	MobileNet version	Dataset environment	Reported performance metrics
1	Zaki et al. 2020	V2	Controlled	Test accuracy = 95.94%
2	Gunarathna and Rathnayaka 2020	V1	Controlled	Training Accuracy = 91.12% Validation Accuracy = 90.78% Classification accuracy = 87.68%
3	Begum et al. 2020	V1	Controlled	Classification accuracy = 97%, Precision = 0.96, Recall = 0.94, F1-score = 0.95
4	Li et al. 2021	V1	Unspecified	Classification accuracy = 87
5	Proposed method	V3	Controlled and Uncontrolled	Training accuracy = 96.23% Validation accuracy = 98.42% Test accuracy = 85.25% Classification accuracy = 81% Precision = 0.76 Recall = 0.95 F1-score = 0.84

Conclusion

This study used the MobileNetV3 model to detect tomato late blight disease with high precision, recall, F1-score, and classification accuracy. The model was tested on real-world images and has the potential to be integrated into a mobile app for practical use. Future research could adapt the model to detect multiple infections and optimize it using techniques such as LAM. This will potentially improve the precision and accuracy of the model. These results demonstrate the potential for CNNs to be a powerful tool for automating the detection of LB disease in tomato plants and contribute to the field of plant disease diagnosis.

Acknowledgements

The authors would like to thank the Mbeya University of Science and Technology (MUST) for the financial and moral support during this study, Dr David Maleko from Sokoine University of Agriculture (SUA) for his assistance in identifying diseased leaves, and tomato farmers in Mbeya, Morogoro and Arusha for their kind assistance and permission to visit their farms and take photographs.

References

Ashwinkumar S, Rajagopal S, Manimaran V and Jegajothi B 2022 Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Mater. Today: Proc.* 51: 480-487.

Barman U, Sahu D, Barman GG and Das J 2020 Comparative assessment of deep learning to detect the leaf diseases of potato based on data augmentation. In *2020 International Conference on Computational Performance Evaluation (ComPE)*: 682-687. IEEE.

Begum AS, Savitha S, Shahila S, and Sharmila S 2020 Diagnosis of Leaf Disease Using Enhanced Convolutional Neural Network. *Int. J. Innov. Res. Appl. Sci. Eng.* 3(12): 579–586.

Benard R, Frankwell D and Ngalapa H 2014 Assessment of information needs of rice farmers in Tanzania; A case study of Kilombero District, Morogoro. *Library Philosophy and Practice (e-journal).* 1071.

Bi C, Wang J, Duan Y, Fu B, Kang JR and Shi Y 2020 MobileNet based apple leaf diseases identification. *Mobile Networks and Applications*: 1–9.

Chen J, Zhang D, Zeb A and Nanehkaran YA 2021 Identification of rice plant diseases using lightweight attention networks. *Expert Syst. Appl.* 169: 114514.

Costa JM and Heuvelink EP 2018 The global tomato industry. In Tomatoes. 27: 1–26. Boston, USA: CABI.

Dan B, Sun X and Liu L 2019 Diseases and pests identification of Lycium barbarum using se-mobilenet v2 algorithm. In *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*. 1: 121–125. IEEE.

Elfatimi E, Eryigit R and Elfatimi L 2022 Beans leaf diseases classification using mobilenet models. *IEEE Access* 10: 9471–9482.

Francis M and Deisy C 2019 Disease detection and classification in agricultural plants using convolutional neural networks–a visual understanding. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN): 1063-1068. IEEE.

Gunarathna MM and Rathnayaka RMKT 2020 Experimental Determination of CNN Hyper-Parameters for Tomato Disease Detection using Leaf Images. In *2020 2nd International Conference on Advancements in Computing (ICAC)*. 1: 464–469. IEEE.

Howard AG, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V and Le QV 2019 Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 1314–1324.

Howard AG, 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*.

Huang Z, Qin A, Lu J, Menon A and Gao J 2020 Grape Leaf Disease Detection and Classification Using Machine Learning. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. 870-877. IEEE.

Kandel I and Castelli M 2020 The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*. 6(4): 312–315.

Kingma DP and Ba J 2014 Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kulkarni O 2018 Crop disease detection using deep learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. 1–4. IEEE.

Li Z, Liu F, Yang W, Peng S and Zhou J 2021 A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.

Meya AI, Mamiro DP, Kusolwa PM, Maerere AP, Sibuga KP, Erbaugh M, Miller SA and Mtui HD 2014 Management of tomato late blight disease using reduced fungicide spray regimes in Morogoro, Tanzania. *Tanz. J. Agric. Sci.* 13(2): 8–17.

Mukhtar H, Khan MZ, Khan MUG and Younis H 2021 Wheat disease recognition through one-shot learning using fields images. In *2021 International Conference on Artificial Intelligence (ICAI)*. 229–233. IEEE.

Mutayoba V, and Ngaruko D 2018 Assessing tomato farming and marketing among smallholders in high potential agricultural areas of Tanzania. *Int. J. Econ. Commerce Manag*. 577–590.

NBS 2017 Annual Agriculture Sample Survey Crop and Livestock Report. https://www.nbs.go.tz/nbs/takwimu/Agriculture/2016-17_AASS Report _Final.pdf

National Bureau of Statistics(NBS) 2021 The National Sample Census of Agriculture 2019/20. Retrieved from https://www.nbs.go.tz/index.php/en/census-surveys/agriculture-statistics/659-2019-20-national-sample-census-of-agriculture-key-findings-report

Nelson SC 2008 Late blight of tomato (*Phytophthora infestans*). Plant Disease PD-45, Aug. 2008

Nguyen TV and Paik I 2020 Feature Extraction with Triplet Loss to Classify Disease on Leaf Data. In *2020 11th International Conference on Awareness Science and Technology (iCAST)*.1–5. IEEE.

Nowicki M, Kozik EU and Foolad MR 2013 Late blight of tomato. *Translational Genomics for Crop Breeding: Biotic Stress*. 1: 241–265.

Qian S, Ning C and Hu Y 2021 MobileNetV3 for image classification. In 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). 490–497. IEEE.

Rajbongshi A, Sarker T, Ahamad MM and Rahman MM 2020 Rose diseases recognition using MobileNet. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. 1–7. IEEE.

Sankaran S, Mishra A, Ehsani R and Davis C 2010 A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* 72(1): 1–13.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1): 1929–1958.

Xie C, Shao Y, Li X and He Y 2015 Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging. *Sci. Rep.* 5: 1–11.

Zaki SZM, Zulkifley MA, Stofa MM, Kamari NAM and Mohamed NA 2020 Classification of tomato leaf diseases using MobileNet v2. *IAES Int. J. Artif. Intell.* 9(2): 290.