Open Journal of Physical Science (OJPS)
ISSN: 2734-2123
Article Details:
DOI: 10.52417/ojps.v5i1.585
Article Ref. No.: OJPS0501002-585
Volume: 5; Issue: 1, Pages: 18- 31 (2024)
Accepted Date: 4th November, 2024

Open Journals Nigeria (OJN)
Open Access | Bi-annual | Peer-reviewed
www.openjournalsnigeria.org.ng
editorial@openjournalsnigeria.org.ng

RESEARCH ARTICLE **OJPS0501002-585**

A COMPREHENSIVE EVALUATION OF MOBILENET ARCHITECTURE FOR TOMATO DISEASES.

Abdullahi, T., *George, G. & Shehu, A.

Department of Computer Science, Faculty of Computing and Applied Sciences, Baze University, Jabi Abuja, Nigeria

***Corresponding Author Email:** gilbertgeorge@bazeuniversity.edu.ng*

ABSTRACT

The potential of deep learning models to automate and enhance various processes has garnered significant attention for their use in agricultural applications in recent years. One notable application is the use of convolutional neural networks (CNNs) for classifying plant diseases. An extensive assessment of the MobileNet architecture for the task of classifying tomato diseases is presented in this research. Because of its lightweight architecture, MobileNet is renowned for its effectiveness and adaptability for embedded and mobile devices. We use a publicly available dataset to investigate MobileNet's effectiveness in classifying various tomato illnesses. Comparing MobileNet to other deeper topologies, experimental results show how successful it is at achieving high accuracy with reduced computational complexity. We obtained 97% accuracy, classifying nine disease categories plus one healthy category using the leaves of the tomato plant as a feature.

Keywords: *MobileNet, tomato disease classification, Deep learning, Convolutional neural networks, Agricultural applications.*

INTRODUCTION

Agriculture has always been associated with the cultivation of key crops deemed necessary for our diet and, more importantly, our survival. It is one of the basic human necessities. Agriculture research mainly tries to improve food productivity and quality. Livestock, fishing, forestry, and agricultural production are the four main industries of agriculture in a developing nation like Nigeria. With crop production accounting for about 87.6% of total output, it is the most prominent sector (Taiwo Oyaniran, 2020). The remaining industries provide 8.1%, 3.2%, and 1.1% of the total for forestry, fishery, and livestock, respectively. Agriculture contributes around 24% to Nigeria's GDP, making it one of the largest sectors in the country. It is essential for both domestic consumption and export (National Bureau of Statistics, 2024). Moreover, this sector is the largest in the country, employing over 36% of the labor force. Because of this, the governments of many developing nations dedicate a sizable portion of their yearly budgets to advancing agriculture and implementing scientific and technological farming methods (Sarfo et al., 2024).

Tomato is one of the crops highly valued in Nigeria, especially for preparing rice and stew, and it is a very short-duration crop with high production and commercial worth (Ugonna *et al.*, 2015). Tomatoes are fleshy vegetable fruits that can be cultivated in small plots in the garden or in huge commercial numbers. It is a popular element in Nigerian cuisine(Danmaigoro et al., 2024). In 2023, Nigeria's total tomato production was approximately 3.7 million metric tons. This figure reflects a steady increase in the country's tomato output, driven by improvements in farming practices and a focus on increasing agricultural productivity. Nigeria ranks among the top 10 tomato-producing countries globally, contributing significantly to the overall supply of fresh tomatoes in Africa (Danmaigoro et al., 2024).

According to (Damicone *et al.*, 2007), tomato infections are divided into two categories. The initial type of infection arises from pathogenic microorganisms, including fungi, bacteria, viruses, and nematodes (Chandrashekar & George, 2024). The second group includes non-infectious factors related to physical or chemical causes, such as adverse environmental conditions, nutritional or physiological irregularities, and damage from herbicides. Tomato problems can arise from bacteria, fungi, or improper cultivation practices, with bacteria responsible for 16 diseases and insects causing five additional issues. Ralstonia solanacearum is the bacteria that causes a severe case of bacterial wilt. This bacterium can live in the soil for a very long time and enter roots through wounds caused by insects, farming, transplanting, or other human activity. It can also enter roots naturally by secondary root emergence. High moisture content and high temperatures are ideal for the growth of diseases. The bacteria rapidly multiply within the plant's water-conducting tissues, leading to the accumulation of slime. The vascular system of the plant is impacted, even though the leaves might not turn yellow(Hari et al., 2024). An infected plant's cross-section shows a brown stem with a substance that seems yellowish seeping out (Damicone *et al.*, 2007).

Currently, professionals rely on manual inspection to detect plant illnesses. This method requires constant monitoring by numerous experts, making it costly for large farms (Singh *et al.*, 2015). Additionally, farmers in some countries are not proactive in seeking expert assistance. Visual detection of plant diseases through physical inspection is a labor-intensive process that is often less precise and limited in scope. The vascular system of the plant may be affected by this technique, even if the leaves might stay green. There is a need for automated, **cost-**effective, and scalable solutions that can help farmers detect and manage tomato diseases early and accurately.

The aim of this study is to provide a comprehensive evaluation of the MobileNet architecture in classifying tomato diseases using image data, with a focus on practical deployment in real-time agricultural applications. This research seeks to determine the accuracy, efficiency, and practicality of MobileNet for real-time disease diagnosis, contributing to the advancement of agricultural technologies that can help farmers identify and manage crop diseases more effectively.

The objectives of this study were to:

i. review existing methods for tomato disease classification, with a focus on deep learning models and their performance in agricultural applications.

ii. implement and fine-tune MobileNet for the classification of common tomato diseases, utilizing a dataset of labeled images.

iii. assess the accuracy and efficiency of the MobileNet architecture in identifying various tomato diseases compared to other convolutional neural network (CNN) models.

iv. propose enhancements or modifications to MobileNet (if needed) to improve classification accuracy, especially in challenging scenarios such as partial occlusion or poor lighting conditions.

v. evaluate the potential for real-time deployment of the model in practical farming applications, focusing on ease of use and integration with existing agricultural tools or apps.

a. CONVOLUTIONAL NEURAL NETWORK

The feature extraction process, a linear operation called convolution plays a crucial role. This operation entails applying a small array of numbers, known as a kernel or filter, to an input tensor array. The kernel slides over the input, computing dot products at each position. These dot products capture local patterns and features, allowing the network to learn relevant representations. Convolutional neural networks (CNNs) heavily rely on this convolution operation to extract meaningful features from images, text, or other data. The resulting feature maps serve as input to subsequent layers, enabling hierarchical feature learning. Convolutional networks have positioned deep learning as the leader in nearly every machine learning task. After a brief mathematical introduction to convolution, its primary use in CNNs is to extract features from input images (Christopher et al., 2023; Shin et al., 2016).

For high-resolution images, artificial neural networks (ANNs) struggle with the curse of dimensionality, whereas convolutional neural networks (CNNs) perform some pre-processing. This allows the network to learn from filters before making the final classification. By enforcing pattern connectivity among neurons in adjacent layers, CNNs use filters to detect spatial locality. Image processing involves multiplying each element in a matrix by its neighboring elements, with the kernel (or filter) weighting or biasing the results.

In their study, Zhang *et al.* (2018) employed a deep convolutional neural network (CNN) for tomato leaf disease identification using transfer learning. The CNN backbone included AlexNet, GoogLeNet, and ResNet. The best-combined model was modified to examine the performance of training and fine-tuning the CNN. Experimental results demonstrated that the proposed technique effectively identifies tomato leaf disease and can be generalized to detect other plant diseases. The storage space required by the proposed model is approximately 1.5 MB, whereas the pre-trained models used in the study require around 100 MB, highlighting the efficiency of the proposed model over pre-trained ones.

Gupta (2020) developed an automatic technique that uses the unique features of diseased and healthy leaves to identify tomato plant leaf diseases. This approach entailed comparing the output of the neural network across different data subsets while employing a back-propagation neural network (BPNN) for pattern identification. This procedure consists of the following steps: gathering images, pre-processing, feature extraction, creating subsets, and classification. The study examined five different kinds of diseases that affect tomato leaves, primarily concentrating on using CNN techniques to identify and categorize these diseases.

To identify and categorize tomato leaf diseases, Ahmad *et al.* (2020) evaluated twenty-three CNN architectures, including VGG-19, VGG-16, ResNet, and Inception V3, through feature extraction and parameter tuning. They tested the models using two datasets: a lab-based dataset and a self-collected field dataset, each with five different disease categories. The lab-based dataset generally produced better results, with performance variations of 10% to 15% across several parameters compared to the field dataset. Their main goal was to develop the most effective machine-learning model for identifying tomato crop diseases in standard RGB images (Ouhami *et al.*, 2020).

Deep learning models incorporating transfer learning, including DenseNet 121 and VGG16, were investigated as potential solutions to this problem. The study concentrated on photos of diseased plant leaves that were categorized into six categories of plant illnesses and pest infestations. A pre-trained deep convolutional neural network (CNN) was the foundation for the plant leaf disease diagnostic model that Anandhakrishnan and Jaisakthi (2020) suggested. Ten different tomato leaf classes from an available dataset were used to train the model. By experimenting with different batch sizes, dropout rates, and training epochs, the model's performance was optimized. Even though the architecture performed well, more study is required to speed up computer time spent using photos to diagnose plant illnesses.

A system for categorizing apple illnesses according to color, texture, and shape features was created by Dubey and Jalal (2015). These features are computed, contaminated fruit segments are identified using K-means clustering, and apples are classified as healthy or contaminated using a multi-class support vector machine (SVM). The method reaches up to 93% accuracy.

Three deep learning meta-architectures were implemented for plant disease identification by Seleem *et al.* (2020) using the TensorFlow object detection framework: the Single Shot MultiBox Detector (SSD), Region-based Fully Convolutional Networks (RFCN), and Faster Region-based Convolutional Neural Network (RCNN). These models were trained and assessed using a dataset from a controlled environment. To improve the mean average precision (mAP) of the best-performing architecture, many advanced deep-learning optimizers were applied. With an average mean precision of 73.07%, the SSD model—which was trained using the Adam optimizer—performed the best.

MATERIALS AND METHOD

This section details the CNNs used for classifying tomato plant leaf diseases, focusing on developing the most suitable CNN model. The entire process is divided into essential steps, as illustrated in Figure 1. The system design is presented in a flowchart, covering the stages of image acquisition, image enhancement and color space modification, image segmentation, feature extraction, and the classification task.

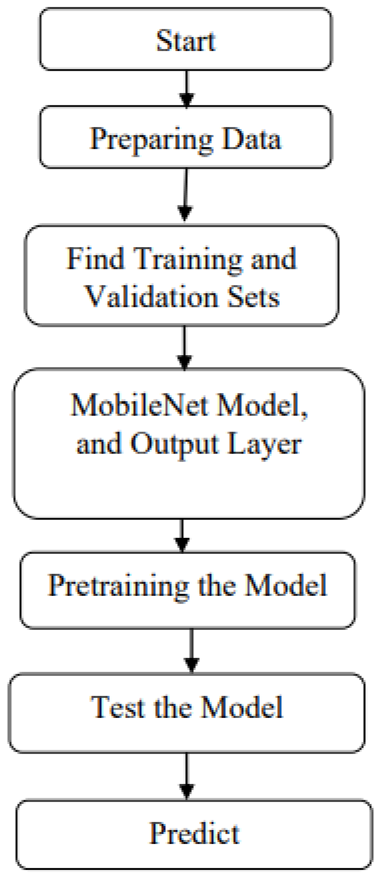

Figure 1: Research Design.

a. In the data preparation stage, we gather a dataset for input. For image data, we scale and normalize the pixel values (ranging from 0 to 255) by dividing them by 255. The input photographs are resized to 224x224 pixels with three channels (Red, Green, and Blue). During feature learning, we apply convolution and pooling procedures. The dataset, sourced from Kaggle, is used for training and validation sets. We employ the MobileNet model and perform output layer pretraining. Testing and prediction datasets are captured using a digital camera on a farm in Afaka, Igabi (LGA) of Kaduna State. The photos are taken in an uncontrolled setting to maintain their original hue. With the guidance of Dr. H.A. Hamisu from the College of Agriculture

(ABU) in Mado Kaduna provides. The research project spans five weeks, monitoring tomato leaves to avoid color effects from chemical products. The farm is located in Kaduna State's Mando, Igabi (LGA), and the Kaggle dataset includes images of diseased tomato leaves categorized the disease classes include bacterial Spot, early blight, late blight, leaf mould, septoria leaf mould, spider mites, two-spotted spider mite, target spot, yellow leaf curl virus, and healthy leaves.

b. **Testing and Validation**: Before the model preparation, the data set is divided into training and validation. The CNN model is trained on a dataset kept for training and validated by the validation set. The dataset was divided into training with 80% and validation having 20% to avoid under-sampling from occurring.

c. **MobileNet Model and Output Layer**: A series of layers defines the defined model in Keras. We employed an integrated deep learning system that uses an ensemble model for detection and classification, a pre-trained MobileNet model for feature extraction, and stacking for detection. Originally introduced in (SIfre & Mallat, 2014), the MobileNet model is primarily built from depth-wise separable convolutions. It was then utilized in the Inception models (Chollet, 2017) to reduce the amount of computation constraints in the initial layers. The suggested Mobilenet architecture is based on the fundamental Mobilenet paradigm. It can be described as a Mobilenet model that has been trained architecturally. There are enormous amounts of neurons in hidden levels. Before sending the input images to the next layer, the neurons in the hidden layer perform conversions by determining the input image's length, width, and height. The weights are modified to become more predictive of image class as the network learns. The pre-trained CNN Model, or MobileNet, is the model that is employed. Lastly, the strength or amplitude of a connection between 28 neurons is the expected property of the Output Layer Neuron Weights. Unit of Rectified Linear Functions are used in regression equations to compare weights on inputs such as coefficients. Every input node has its weights initialized, usually at random, with a tiny number in the 0–1 range. Among the most often used learning algorithms are feed-forward supervised neural networks, also known as multilayer perceptions or Deep Networks. In a single hidden layer, every neuron is connected to other neurons with varying degrees of weight. Neurons in each hidden layer would be activated by the upward network progressions of the input layers, leading to the output value at the end. We refer to this type of network as feedforwarding. Disabling down sample layers, SoftMax activation function, heterogeneous kernel-based convolutions, and augmentation are some of the changes that are included. High probability is obtained by using the SoftMax activation function for the illness classes' output.

d. As image data flows through this deep neural network, it encounters multiple hidden layers. These layers process features, remove outliers, identify significant patterns, and produce the predicted output. By removing the last six layers, disabling sample layers, and applying heterogeneous kernel-based convolutions, MobileNet reduces its model size. Additionally, using the SoftMax activation function and augmentation techniques enhances MobileNet's accuracy. Collectively, these changes constitute the proposed MobileNet model, which is explained in detail in this section.

e. **Pretraining the Model**: The model is compiled using efficient numerical libraries located in the backend, such as TensorFlow or Theano. The backend selects the optimal method for representing the network during training and generates predictions for the hardware. Initially, the model had few layers, resulting in underfitting and lower accuracy in predicting leaf types. For training, we use a predetermined number of epochs (25) and a relatively small batch size (25). The SoftMax activation function is influenced by the Dense layer. We apply the Adam algorithm with a learning rate of 0.0001, using 'categorical_crossentropy' loss and accuracy metrics. Finally, we begin training with 25 epochs and 40 steps per epoch. Verification data is then used to test the trained model against the valid_generator dataset and determine its accuracy.

f. **Testing the Model**: After training the neural network on the training and validation datasets, we evaluate the model's performance. However, this evaluation on the same dataset does not provide insight into how the model will perform on new data. To assess its generalization ability, we use the test dataset. Evaluation metrics serve to assess the classifier's performance. By comparing the model's predictions with actual values from the test dataset, we apply mathematical methods. The precision metric calculates the percentage difference between accurately identified samples and the total number of instances. Accuracy is obtained by the ratio of correct classifications to the total number of categories as seen in equation 1 (Chollet, 2018).

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (1)$$

g. **Dataset**: A dataset comprising 4,860 tomato leaf images was collected for training, while 1,620 images were allocated for validation, and 323 images were used for testing. All images were stored in JPG format. The dataset includes nine disease classes related to tomato leaves, such as bacterial spot, early blight, late blight, leaf mould, Septoria leaf mould, spider mites, two-spotted spider mites, target spot, yellow leaf curl virus, and healthy leaves. The model's dataset was divided into 80% training and 20% validation portions. Finally, 323 image samples were used to evaluate the model's effectiveness.

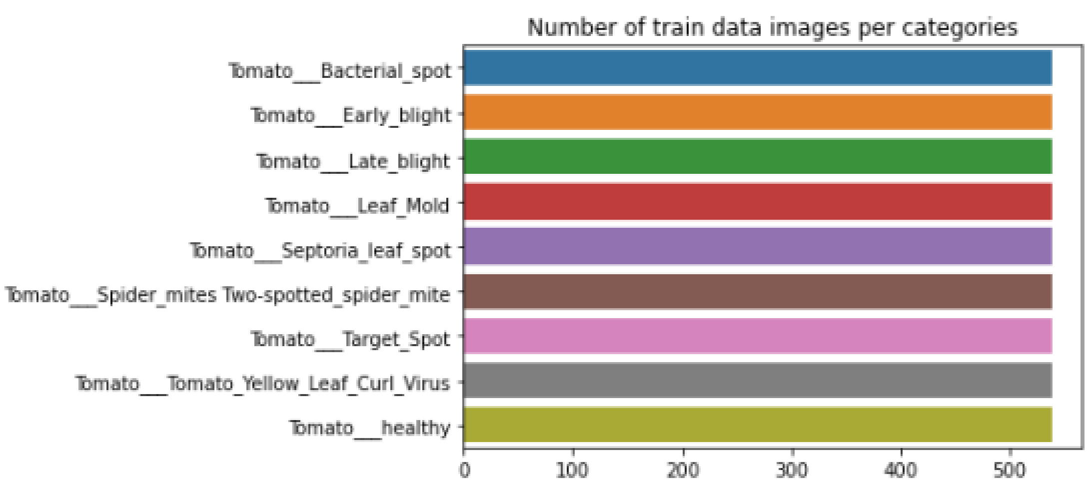

Figure 2: number images per category

h. **Environment**: A Computer with a CPU (core I5, 2.6GHz) is used to conduct model training. Tools required for this work have been installed. Accordingly, python version 3.7.4 was the one installed. All models built in work underwent similar data preparation processes. The difference in the models exists in the concept of architecture and the technique used to achieve the aim.

RESULTS AND DISCUSSION

A. TRAINING RESULT

In the defined architecture, a MobileNet architecture is employed using a linear stack of layers. The model is compiled with specified loss functions, optimizers, and metrics. It takes image input (e.g., photographs) and produces relevant outputs. One-hot coding contrasts categorical variables, transforming them into binary variables. Challenges include irregular image shapes and potential overfitting due to dataset size. Data augmentation techniques enhance image classification capabilities. Dropout layers are applied to improve accuracy. Training details involve SoftMax activation, the Adam algorithm, and evaluation using a validation dataset.

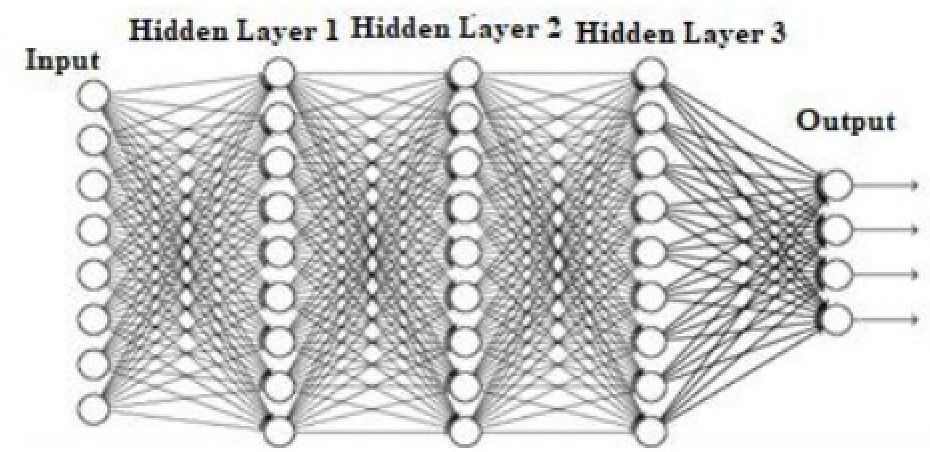

Figure 3: CNN architecture

```
40/40 - 646s - loss: 0.0763 - accuracy: 0.9758 - val_loss: 0.2431 - val_accuracy: 0.9203
Epoch 9/25
40/40 - 673s - loss: 0.0707 - accuracy: 0.9789 - val_loss: 0.1478 - val_accuracy: 0.9578
Epoch 10/25
40/40 - 749s - loss: 0.0641 - accuracy: 0.9812 - val_loss: 0.1354 - val_accuracy: 0.9594
Epoch 11/25
40/40 - 486s - loss: 0.0477 - accuracy: 0.9867 - val_loss: 0.1313 - val_accuracy: 0.9563
Epoch 12/25
40/40 - 482s - loss: 0.0490 - accuracy: 0.9812 - val_loss: 0.1223 - val_accuracy: 0.9586
Epoch 13/25
40/40 - 482s - loss: 0.0456 - accuracy: 0.9867 - val_loss: 0.1466 - val_accuracy: 0.9508
Epoch 14/25
40/40 - 486s - loss: 0.0324 - accuracy: 0.9914 - val_loss: 0.1272 - val_accuracy: 0.9563
Epoch 15/25
40/40 - 486s - loss: 0.0478 - accuracy: 0.9875 - val_loss: 0.1498 - val_accuracy: 0.9625
Epoch 16/25
40/40 - 488s - loss: 0.0435 - accuracy: 0.9844 - val_loss: 0.1438 - val_accuracy: 0.9594
Epoch 17/25
40/40 - 490s - loss: 0.0345 - accuracy: 0.9891 - val_loss: 0.1322 - val_accuracy: 0.9641
Epoch 18/25
40/40 - 488s - loss: 0.0371 - accuracy: 0.9891 - val_loss: 0.1351 - val_accuracy: 0.9617
Epoch 19/25
40/40 - 490s - loss: 0.0355 - accuracy: 0.9891 - val_loss: 0.1058 - val_accuracy: 0.9727
Epoch 20/25
40/40 - 490s - loss: 0.0286 - accuracy: 0.9930 - val_loss: 0.1123 - val_accuracy: 0.9648
Epoch 21/25
40/40 - 493s - loss: 0.0199 - accuracy: 0.9953 - val_loss: 0.1061 - val_accuracy: 0.9648
Epoch 22/25
40/40 - 483s - loss: 0.0273 - accuracy: 0.9906 - val_loss: 0.0805 - val_accuracy: 0.9750
Epoch 23/25
40/40 - 481s - loss: 0.0142 - accuracy: 0.9969 - val_loss: 0.0801 - val_accuracy: 0.9734
Epoch 24/25
40/40 - 481s - loss: 0.0270 - accuracy: 0.9914 - val_loss: 0.1070 - val_accuracy: 0.9680
Epoch 25/25
40/40 - 485s - loss: 0.0238 - accuracy: 0.9922 - val_loss: 0.0760 - val_accuracy: 0.9711
```

Figure 4: Training and Validation process

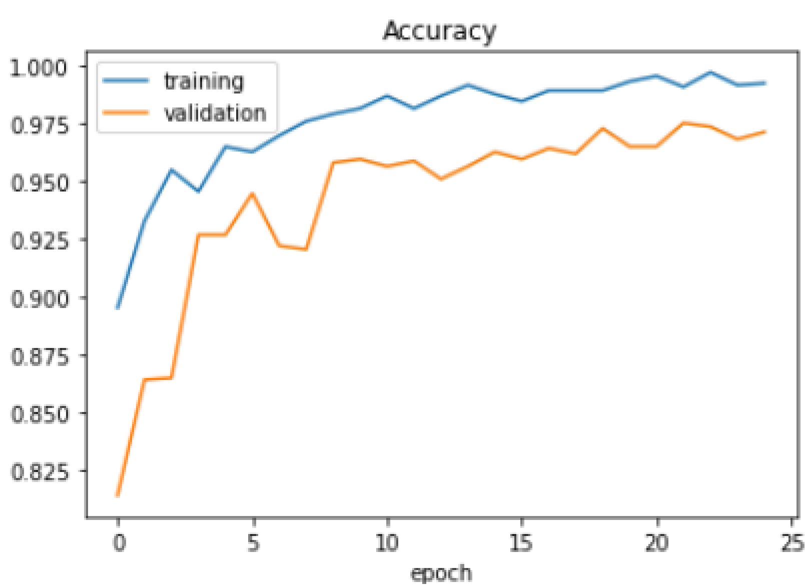

Figure 5: Accuracy chart per epoch

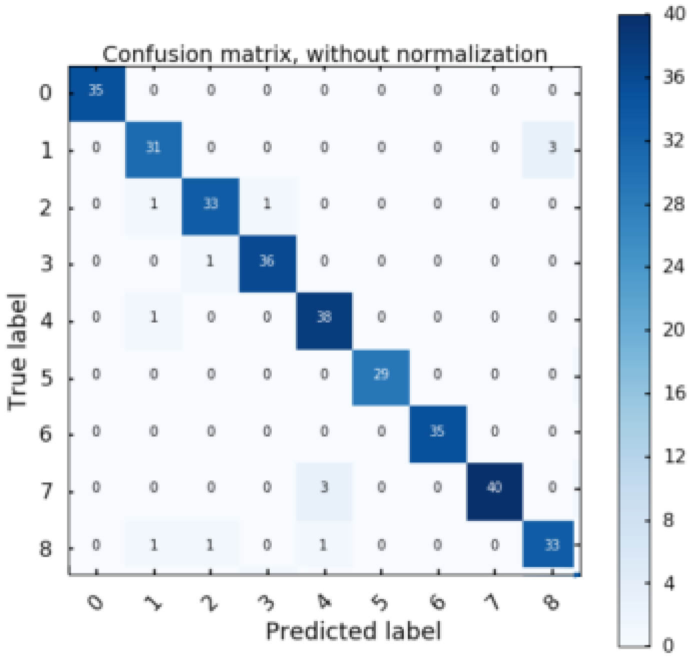

Figure 6: Confusion matrix

B. TEST RESULTS

After training our neural networks on the training and validation data sets, the model is now being evaluated by using the test data to verify the network's performance on images it has not seen throughout the training epoch; the prediction was able to score 97.1% as shown in Figure 6.

```
#Model Accuracy
int_equivalent=(evaluate[1])
approx_acc=round(int_equivalent, 3)
str_equi =str(approx_acc* 100)
print('Model accuracy:'+str_equi+'%')
```

```
Model accuracy:97.1%
```

Figure 7: Evaluation result on the test dataset

The training results show the effectiveness of the chosen MobileNet architecture in image classification tasks, highlighting both its strengths and areas for potential improvement. Through the application of a linear stack of layers, MobileNet achieves efficient processing with relatively low computational requirements, which is ideal for image input processing. The model's configuration, featuring specific loss functions, optimizers, and metrics has been optimized to enhance learning accuracy and reduce classification errors. One-hot encoding effectively prepares categorical data, allowing the model to differentiate between classes in a clear binary format, which is essential for accurate label classification.

Challenges encountered during training include handling images of varying shapes, which can hinder consistency across the dataset. To address this, data augmentation techniques such as random rotations, cropping, and scaling were applied, which not only standardize inputs but also improve the model's robustness. This augmentation approach, coupled with the addition of dropout layers, mitigates overfitting by introducing regularization, thereby enhancing generalizability. The use of SoftMax activation and the Adam optimizer further supports stable learning, while the validation dataset enables ongoing monitoring of the model's performance.

Upon testing with unseen data, the model achieved a high accuracy of 97.1%, as illustrated in Figure 6's confusion matrix, which visually confirms its ability to accurately classify most images. This impressive performance demonstrates that the model's training parameters and structure are well-suited to the task, and it effectively generalizes to new data. Future work could involve exploring additional regularization techniques or experimenting with more complex architectures to improve upon the already strong results, particularly if addressing larger or more complex image datasets.

CONCLUSION

In our proposed task, we trained a MobileNet-based CNN model to detect tomato crop diseases. Specifically, we focused on the last six layers of MobileNet and adjusted the dense layer to accommodate 46 input channels using the original weights. For experimentation, we utilized tomato data from the Kaggle dataset, which includes nine disease categories plus a healthy category. Notably, the dataset maintains a balanced distribution of healthy leaf category images. To enhance model accuracy, we applied data augmentation techniques. For predictions, we evaluated the model's performance using a locally collected dataset. Our implementation leveraged Python programming, TensorFlow, and Keras libraries within the Anaconda environment, achieving an impressive accuracy score of 97.1%.

Contribution to Knowledge: The contributions of our research are to improve the quality and quantity of tomato production by using image processing techniques to distinguish disease classes from healthy leaves. This enables more accurate detection and classification of diseases potentially harmful to the plant throughout its lifecycle, allowing farmers to identify types of plant disease more accurately without requiring a crop scientist to determine the kind of crop disease present in the farm or garden. Since not all farmers can afford constant checkups from crop scientists, and many cannot identify diseases correctly, this model has the potential to improve tomato production outcomes in Nigeria.

Recommendation: This article's main objective is to develop a pre-trained model to classify the tomato plant's leaf to detect that crop's disease. Future improvements could include detecting diseases in the crop's other components, such as roots, stems, and branches which could further increase the detection accuracy more than the contemporaneous one. Additionally, image categorization will be done to determine whether the leaf belongs to the preferred category or not. Our future objective is to optimize the accuracy and collect vast data to make our dataset on profuse plant diseases. As agriculture employs more than half of our population, this model could serve as a significant resource for farmers. Finally, the pre-trained Mobile Net model was created and can be deployed to the web to allow remote access through the phones and other devices.

CONFLICT-OF-INTEREST DISCLOSURE

No conflict of interest registered.

REFERENCES

Ahmad, I., Hamid, M., Yousaf, S., Shah, S.T., & Ahmad, M.O. (2020). Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection. Complex., 2020, 8812019:1- 8812019:6.

Anandhakrishnan, T., & Jaisakthi, S. M. (2020). Identification of tomato leaf disease detection using pretrained deep convolutional neural network models. *Scalable Computing*, **21**(4): 625–635. https://doi.org/10.12694:/scpe.v21i4.1780

Ansari, S. (2020). Building Computer Vision Applications Using Artificial Neural Networks. In Building Computer Vision Applications Using Artificial Neural Networks. https://doi.org/10.1007/978-1-4842-5887-3

Chandrashekar Uppin and Gilbert George (2024). Assessing the Efficacy of Transfer Learning in Chest X-ray Image Classification for Respiratory Disease Diagnosis: Focus on COVID-19, Lung Opacity, and Viral Pneumonia. *Journal of Image Processing and Artificial Intelligence*, **10**(1): 11-20.

Christopher, M. V., Wahid, A., Nabiilah, G. Z., & Rojali. (2023). Comparing Age Estimation with CNN and EfficientNetV2B1. *Procedia Computer Science*, *227*, 415–421. https://doi.org/10.1016/j.procs.2023.10.541

Christopher, M. V., Wahid, A., Nabiilah, G. Z., & Rojali. (2023). Comparing Age Estimation with CNN and EfficientNetV2B1.Procedia Computer Science, 227, 415–421. https://doi.org/10.1016/j.procs.2023.10.541

Damicone, J., Conway, K., & Brandenberger, L. (2007). Common Diseases of Tomatoes. In: Durmus, H., Gunes, E. O., & Kirci, M., Oklahoma Cooperative Extension Service Epp7625 (Issue Figureure 1).

Danmaigoro, A., Aliyu, A., & Aliyu, A. (2024). Tomato Value Chain in Kebbi State: Issues, Challenges and Strategies. Indonesian Journal of Advanced Research, **3**(9): 1517–1528.

Dubey, S., & Jalal, A. (2015). Apple disease classification using color, texture and shape features from images. *Signal, Image and Video Processing*, **10**(5): 819-826. https://doi.org/10.1007/s11760-015-0821-1

Gupta, R. (2020). Tomato Leaf Disease Detection using Back Propagation Neural Network. *International Journal of Innovative Technology and Exploring Engineering*, **9**(8): 529–538. https://doi.org/10.35940/ijitee.h6531.069820.

Hari, P., Singh, M. P., & Singh, A. K. (2024). An improved federated deep learning for plant leaf disease detection. Multimedia Tools and Applications, 83(35), 83471–83491. https://doi.org/10.1007/s11042-024-18867-9

Ouhami, M., Es-Saady, Y., Hajji, M. El, Hafiane, A., Canals, R., & Yassa, M. El. (2020). Deep transfer learning models for tomato disease detection. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12119 LNCS, 65–73. https://doi.org/10.1007/978-3-030-51935-3_7

Pani, S., Rout, J., Afroz, Z., Dey, M., Sahoo, M.K., Das, A.K. (2023). Diagnosis of Plant Diseases by Image Processing Model for Sustainable Solutions. In: Nandan Mohanty, S., Garcia Diaz, V., Satish Kumar, G.A.E. (eds)

Intelligent Systems and Machine Learning. ICISML 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 471. Springer, Cham. https://doi.org/10.1007/978-3-031-35081-8_15

Saleem, M., Khanchi, S., Potgieter, J., & Arif, K. (2020). Image-Based Plant Disease Identification by Deep Learning Meta-Architectures. *Plants*, **9**(11): 1451. https://doi.org/10.3390/plants9111451

SankarNath, S., & Rakshit, P. (2015). A Survey of Image Processing Techniques for Emphysema Detection. *International Journal of Computer Applications*, **114**(15): 7–13. https://doi.org/10.5120/20052-1983

Sarfo, N. Y., Cornelius, E. W., & Torkpo, S. K. (2024). Greenhouse tomato farmers'knowledge, perceptions, and management of tomato bacterial wilt (*Ralstonia Solanacearum*) disease. *African Journal of Food, Agriculture, Nutrition and Development*, **24**(2): 25537–25558.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Transactions on Medical Imaging, **35**(5): 1285–1298.

Singh, V., Varsha, & Misra, A. K. (2015). Detection of the unhealthy region of plant leaves using image processing and genetic algorithm. Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015, 1028–1032. https://doi.org/10.1109/ICACEA.2015.7164858

Ugonna, C., Jolaoso, M., & Onwualu, A. (2015). Tomato Value Chain in Nigeria: Issues, Challenges and Strategies. *Journal of Scientific Research and Reports,* **7**(7): 501–515. https://doi.org/10.9734/jsrr/2015/16921

Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can Deep Learning Identify Tomato Leaf Disease? Advances in Multimedia, 2018, Article ID: 6710865. https://doi.org/10.1155/2018/6710865