Study on Tomato Disease Classification based on Leaf Image Recognition based on Deep Learning Technology

Ji Zheng*
Guilin University of Electronic Technology
School of Optoelectronic Engineering
541004, China

Minjie Du[2]
Guilin University of Electronic Technology
School of Electronic Engineering and Automation
541004, China

***Abstract*—The utilization of computer vision technology is of the utmost significance in the examination of plant diseases. Research utilizing image processing to investigate plant diseases necessitates the analysis of discernible patterns on plants. Recently, numerous image processing and pattern classification techniques have been employed in the construction of a digital vision system capable of recognizing and categorizing the visual manifestations of plant diseases. Given the abundance of algorithms formulated for the purpose of plant leaf image classification for the detection of plant diseases, it is imperative to assess the accuracy of each algorithm, as well as its potential to identify diverse disease types. The main objective of this study is to explore accurate deep learning architectures that are more effective in deploying and detecting tomato diseases, thus eliminating human error when identifying tomato diseases through visual observation. and get more widespread use. An initial model was constructed from the ground up using a convolutional neural network (CNN), which was trained with 22930 tomato leaf images, and then compared to VGG16, Mobile Net, and Inceptionv3 architectures through a fine-tuning process. The basic CNN model achieved a training accuracy of 90%, whereas the training accuracies of VGG16, Mobile Net, and Inceptionv3 were respectively observed to be 89%, 91%, and 87%. The VGG16 model has a greater computational complexity than other approaches due to its considerable quantity of predefined parameters. Despite to be simpler, MobileNet proved to be the most efficient in terms of accuracy and thus is the most suitable for this research, due to its lightweight structure, fast functioning and adaptability for mobile devices. In contrast to other architectures, the suggested CNN architecture exhibits shallower characteristics, facilitating faster training on the same dataset. This research will provide a solid foundation for future scholars to easily improve the categorization of plant diseases, which is to develop algorithms that are lighter, faster, easier to run, and have higher accuracy.**

Keywords*—Deep learning; convolutional neural network; image recognition; plant diseases

I. Introduction

Agriculture continues to play a significant role in the economy, contributing considerably to foreign exchange earnings and gross domestic product. Tomatoes are a highly sought-after vegetable commodity worldwide due to their rich nutrient content, including organic acids, vitamins, essential amino acids, and natural fiber. It is cultivated both outdoors and indoors. Globally, the annual output of fresh tomatoes is about 160 million tons, which is three times that of potatoes and six times that of rice. In recent years, China's tomato production has steadily increased in terms of cultivated area. According to statistics, China's tomato cultivation area will increase by 1.6% to 1.104 million hectares in 2020. China's tomato cultivation area will increase by 0.7% to 1.113 million hectares in 2021. China is the world's largest producer of tomatoes, with perennial production accounting for as much as one-third of the global total. According to statistics, in 2020 China produced approximately 65.15 million tons of tomatoes, an increase of 3.63 percent. China produces approximately 66.09 million tons of tomatoes in 2021, an increase of 1.44 percent. Nonetheless, numerous maladies on tomato plants cause substantial harm to the quality and yield of tomatoes.[1] Pathogens including bacteria, fungi, and viruses are responsible for producing disease in plants and can be transmitted through soil, water, and air.[2] The use of pesticides and climate change, which can alter the stage and rate of pathogen development, complicate the management of diseases.[3] Common tomato plant maladies include leaf blight, early blight, late blight, target spot, Septoria leaf spot, and yellow leaf curl virus.[4-5] These diseases can affect the leaves, stems, fruits, and roots of the entire tomato plant. If diseases are not controlled in proper time, they will cause significant losses and injury to the tomato plant. Every year, producers incur significant losses. Protecting tomato plants from these maladies is therefore essential for increasing yields. Traditionally, plant diseases are identified through visual observation with the assistance of trained personnel [6-7].

Numerous advancements have been made in computer vision technology in recent years, and there are more examples of applications in identifying and classifying the diseases of fruits and vegetable. Traditional computer vision techniques still require professional knowledge of these agricultural hazards. The development of artificial intelligence technology eliminates the requirement for professionals to conduct constant field observation. [8-9]. It also serves to minimize the possibility of human observational errors. Recent case studies demonstrate that soft computing models like neural networks [10], decision trees [11], support vector machines [12], and Nave Bayes [13] have been applied to the automatic classification of plant diseases. In recent years, deep learning has dominated computer vision. The proliferation of smartphones, the integration of high-performance processors into cameras and mobile devices, and the developments of computer vision technology, particularly with the application of deep learning strategies, have facilitated the automation of diseases diagnosis utilising image recognition. [14]. Deep neural networks are a combination of deep learning and neural networks that replicate the brain's general operating principles. A deep neural network typically comprises an input layer, an output layer, and multiple hidden layers in between. A neural network with a single hidden layer is considered shallow in comparison.

In this paper, the input layer is trained to discover the optimal filter weight values [15]. The initial layer of a CNN model that takes in pictures as input is referred to as the inputted layer. Image height, width, and depth are fed into the input layer of the neural network. The depth indicates how many color channels the image belongs to. If the picture is an RGB picture, for example, it comprises three color components: red, green, and blue, thus the size of the picture is 3. However, if it is grayscale, its size is 1.

In between the input and output layers are the hidden layers. They will be used to identify particular characteristics. The undisclosed layer is used in this research to detect color and texture features that can be used for the categorization of tomato plant maladies. The output layer comprises category labels and has full connection. If we wish to classify an image into ten categories, we must designate ten labels in the fully connected layer. Convolutional neural networks have been revolutionary in several pattern recognition-related disciplines over the past decade [16]. Various deep learning architectures, such as VGG16, Inceptionv3, Mobile Net, Alex Net, Dense Net, and Google Net, have been proposed and all feature slight differences in their respective hidden layers. [17]. Despite all of these learning architectures, the classification of plant maladies remains imperfect.

Currently, CNN has been proven to be a very good choice for achieving plant disease detection in terms of accuracy [18]. Each structure has its limitations. An illustration of this is that certain architectures can be more intricate because a greater number of parameters require more time for processing, yet aid in generating a high degree of precision. The primary objective of this research is to ascertain the most effective classification technique of plant diseases with a view to enhancing this domain further. This is not only in terms of accuracy, but also in terms of speed and difficulty of deploying and adjusting.

However, many plant disease detection methods proposed are not to replace the current detection methods, but to supplement these methods [19]. Despite the promising findings of multiple research studies, the practical implementation of these technologies has yet to be adequately explored. There has been a proliferation of computer-based applications that have demonstrated great efficacy, yet the efficacy of these architectures has not been properly appraised. The primary objective of this research is to analyse the efficacy of modern deep learning systems in regard to plant disease classification with the intent of attaining the highest level of accuracy. This research seeks to provide a definitive answer, rather than a hypothesis, that can be used to assist novice researchers in selecting a current deep learning algorithm that is most advantageous for their initiatives in the field of automated classification of plant diseases.

II. MATERIALS AND METHODS

A. *Methodology Overview*

The methodology implemented in the research is shown in Fig. 1. The process of constructing a model can be divided into five distinct stages: data collection, in which images are acquired and divided into the three primary folders of training, verification and testing; pre-processing, to guarantee that relevant traits are present; the selection of an appropriate architecture; the training and verification stage, in which the model is trained and verified to generate the final model; and the prediction stage. This stage furnishes the classifier with characteristic test images to assess its practical efficacy. The ensuing text contains an elucidation of each phase.

B. *Data Collection*

Ten different categories of 256x256 resolution images of tomato leaves affected by various diseases were procured from the Plant Village Library located on the Kaggle platform. The dataset is segregated into three distinct subsections, namely training, validation, and testing, which contain 17184, 4585, and 1161 images respectively. The utility of testing and verifying using two distinct datasets is to confirm the model's capacity to process new data. The test data is utilised to assess the efficacy of the ultimate model, whereas the verification data is employed to assess the performance of the model during the training phase. Table I illustrates the total amount of tomato leaf images that have been subjected to training, testing, and validation, and classified accordingly. As hardware, the ACER Aspire3 computer uses Intel core TM i5-8265U 1.66Hz, 8GB DDR4 memory, and NVIDIA GeForce MX230 for this research.

In this paper, the method of classifying diseased tomato leaf images focuses on automatically identifying the marks appearing as spots on the leaves, as shown in Fig. 2. Each picture represents a tomato leaf image that belongs to a certain category.

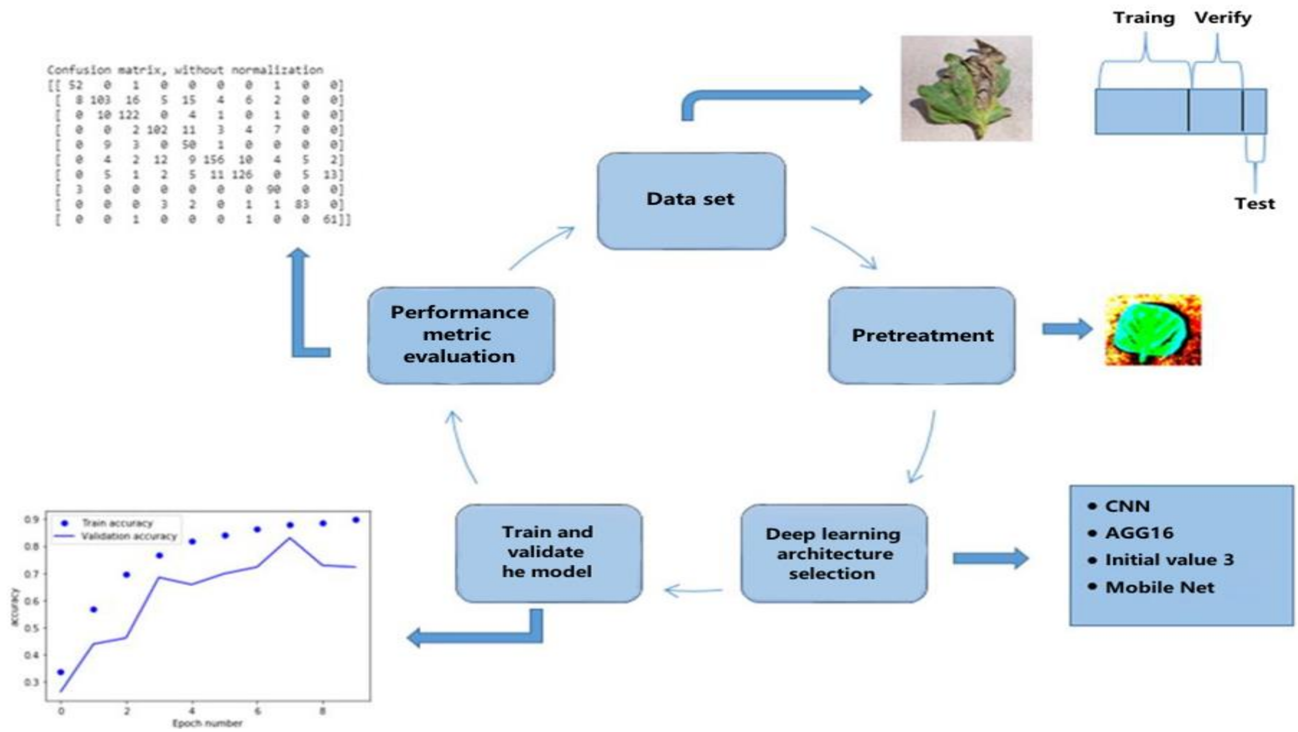

Fig. 1. Steps followed in the research method.

TABLE I. TOMATO LEAF IMAGES OBTAINED DURING THE STUDY

Disease type	Training data set	Validation data set	Test data set
Fine plaque	1684	425	54
Early blight	1761	480	159
Late blight	1713	463	138
Leaf mold	1753	470	129
Septicemic plaque	1682	436	63
Tetranychus	1537	435	204
Target spot	1659	457	168
Mosaic virus	1700	448	90
Yellow leaf curl virus	1868	490	93
Healthy leaves	1863	481	63
Total	17184	4585	1161

C. Image Preprocessing

Image preprocessing is a technique employed to improve the input image for subsequent manipulation, allowing for the pertinent characteristics to be accentuated for further processing [20]. The utilisation of it aids in the alleviation of numerous issues, such as irradiance effects, illumination, and complications resultant from inadequate contrast. Fig. 3(a) and 3(b) demonstrate the difference in a diseased tomato image before and after pretreatment using the Keas input function, respectively. In the preprocessing stage, data standardization and data enhancement technologies are applied.

Data normalization is an essential step in guaranteeing that each pixel in a picture has a comparable dissemination of information. It facilitates rapid convergence of the network during the training process. To normalize the data for this study, the mean was subtracted from each pixel, and the output was divided by the standard deviation.

Following this step, each pixel was assigned a value between 0 and 1. Image enhancement is essential to the development of an efficient image classifier. To construct an accurate classifier, a large number of unique images is required. It is practically impossible to locate so much information. By employing data augmentation techniques, it is feasible to produce fresh data by transforming the current data. In this study, rotation, cropping, scaling, and horizontal and vertical rotating are used as enhancement options.

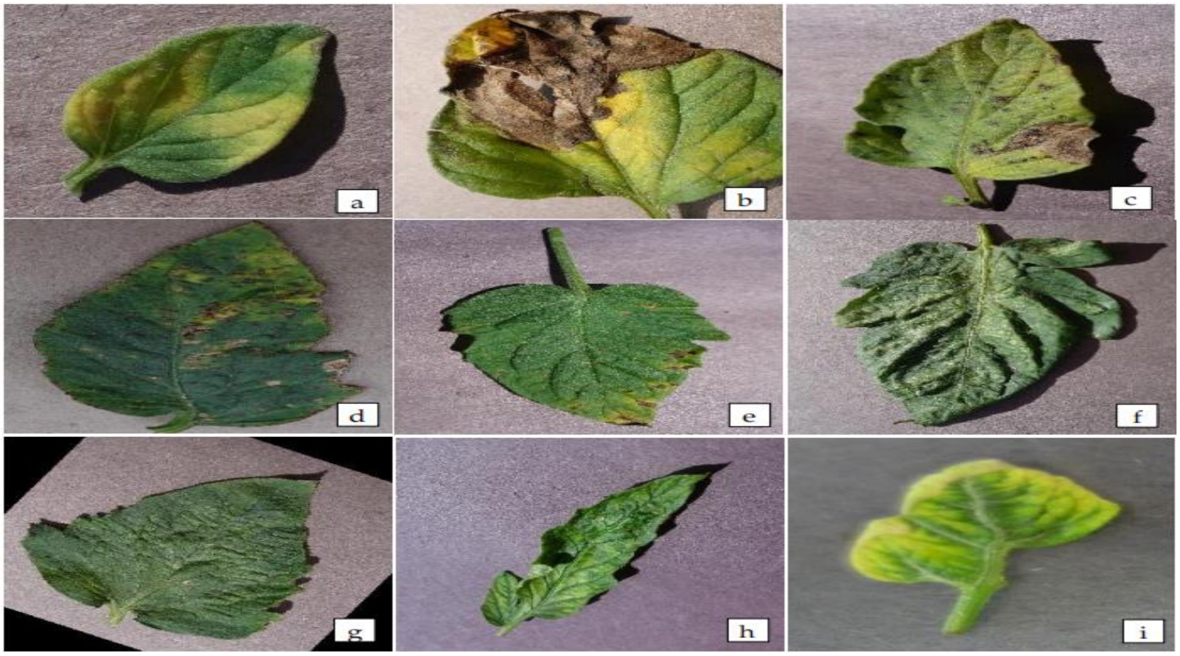

Fig. 2. Image of tomato leaves infected with the following virus, (Note: (a) fine plaque, (b) early blight, (c) late blight, (d) leaf mold, (e) septicemic leaf spot, (f) spider mite, (g) target spot, (h) Mosaic virus, (i) yellow leaf curl virus).

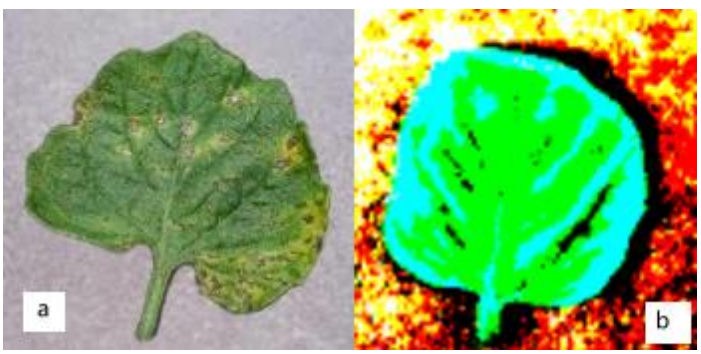

Fig. 3. Images of pretreated tomato leaves (a) before and (b) after.

D. Model Selection

Big data's availability and computational capacity are responsible for the success of deep neural networks. Therefore, it is essential to determine which designs are most effective and under what conditions they can be implemented.

1) Convolutional neural networks *(CNN)*: Convolutional neural network (CNN) is a deep learning model specifically designed to identify and analyse features from multidimensional data sets [21-22]. The selected CNN model consists of four convolutional layers, with subsequent implementation of batch normalization, maximal pooling, and discard layers. Moreover, there are two additional layers, which are dense and flat. Table II presents an overview of the network architecture, comprising of layer configurations and the corresponding number of parameters for training. Convolutional layers have the capability to detect distinct patterns of features, including edges and hues, among other characteristics. If one layer recognizes border patterns, the second layer will recognize line patterns. Each possesses an abstraction level. Batch normalization resembles the preparatory normalization utilized in this study. The only distinction is that this paper employs a different type of layering. This paper makes the image more appropriate for the input layer by normalizing it during the preprocessing phase. In addition, batch normalization is used to make the image more suited for the hidden layer. Normalization of the activation layer's output is performed by subtracting the mean value and dividing by the standard deviation of the batch, resulting in improved stability and speed of the model. To prevent overfitting, the disintegrating and pooling layers are added [14]. In this model, the maximum pooling layer is utilized to reduce the overall size and thereby dynamically boost the computational performance. Each output is the corresponding data after applying maximum pooling to the inputs. Importantly, the maximum pooling contains only hyperparameters, so gradient descent does not necessitate learning. The dropout layer, as indicated by its name, drops out some neurons during training, but keeps them active during validation. As the model's complexity grows, it tends to overfit by memorizing all the weights, but the dropout layer solves this problem by randomly dropping out a portion of each layer during training.

The use of the Flatten Layer between the Merging Layer and the Dense Layer is generally recommended, as it facilitates the transformation of the Merging Feature Map into a single vector, subsequently allowing its passage to the Dense Layer. For each block, the Rectified Linear Unit (ReLU) activation function is employed, however, finally, after the final dense layer, the SoftMax activation function is implemented. The activation function checks which nodes meet the defined conditions in order to pass through the layer. A fixed learning rate of 0.001 is used in the whole training process. The total quantity of epochs is designated to be ten, and the batch magnitude is established to be twenty-seven. The most advantageous amalgamation of parameter values is ascertained by modifying the super parameter. Once the model has been constructed, it must be compiled. The Adam optimization algorithm and the cross-entropy loss function are employed respectively in the compilation of the model as the optimizer and loss function. The model has been finally trained and stored in a .h5 file format.

TABLE II. OUTLINES THE PROPOSED CNN ARCHITECTURE MODEL

Layer	Output shape	Parameter
Conv2d	(None,254,254,32)	896
Max _pooling2d	(None,127,127,32)	0
Conv2d_1	(None,125,125,64)	18496
Max _pooling2d_1	(None,62,62,64)	0
Conv2d_2	(None,60,60,64)	36928
Max _pooling2d_2	(None,30,30,64)	0
dropout	(None,30,30,64)	0
Conv2d_3	(None,28,28,128)	73856
Max _pooling2d_3	(None,14,14,128)	0
Flatten	(None,25088)	0
Dense	(None,128)	3211392
Activation	(None,128)	0
Dropout_1	(None,128)	0
Dense_1	(None,10)	1290
Activation_1	(None,10)	0

2) VGG16: VGG16 is a famous CNN model, which was put forward by VGG of Oxford University. Compared with Alex Net [23], an improvement of VGG16 is to replace the larger convolution kernel (11x11, 5x5) in Alex Net with several successive 3x3 convolution kernels [24]. In this work, the VGG16 model which had been pre-trained was adjusted for the purpose of plant disease identification. In this study, Transfer Learning and Fine-Tuning are employed to classify diseased leaves, leveraging the fact that the VGG16 model has already learnt the characteristics of plant leaves. Utilizing the preprocessing input feature of Keras VGG16, preprocessing is applied before the image is supplied to the input layer. The VGG16 model contains a total of 138,357,544 parameters, which can be obtained from an online source. Given that the model has assimilated these characteristics, it is not indispensable to alter the weights throughout the entire learning procedure. Consequently, the weights are kept constant to impede modification. The number of parameters in the model has been reduced to 134,301,514 due to the freezing of weights. The ultimate model is in accordance with the Adam optimization algorithm and the cross-entropy loss function for classification. For this model, the learning rate is set to 0.001, the batch size is set to 27 and the number of epochs is set to 10. The model has been successfully trained and the weights have been stored in the .h5 format.

3) MobileNet: Mobile Net is an example of a lightweight and high-speed family of deep neural networks that can be deployed for categorization and pattern identification tasks, analogous to other convolutional neural networks. In spite of its diminutive size when compared to other models, it demonstrates remarkable performance. [25] The model's magnitude is primarily determined by the cumulative number of parameters. This research employed a model that was sourced from the web with the use of the Keras Library and pre-processed with Mobile Net's capabilities, which are commonly found to require fewer data augmentation procedures as compared to other models. After the process of refinement, the final model was composed of 3,239,114 parameters. The sample size was ascertained as 27, the number of epochs was established at 10, and the learning rate was maintained at 0.001. The model which had been trained was subsequently stored in .h5 format in order to be used at a later date.

4) Inceptionv3: Szegedy et al. discovered the initial concepts belonging to the family of deep neural networks [26] Several initial models were introduced, and each was designated initial VN, where N represents the version number.[27] In this investigation, Inceptionv3, the third version in the family, was utilized. A model with 21,802,784 parameters was obtained by utilising the Keras library. In addition to the initial architecture, a layer with global average pooling, a discarding layer, two layers with dense connections, and a densely connected output layer with ten nodes were included. The utilisation of a global average pooling layer facilitates the identification of features, while simultaneously reducing superfluous parameters. The number of epochs, learning rate, and batch size were set to ten (10), 0.001, and twenty-seven (27) respectively. The weights of the bottom 172 layers were kept constant while the higher layers were re-trained. Following the fine-tuning process, the total number of parameters in the final model was determined to be 23,387,434, which is slightly higher than the number of parameters present in the original model. The model is ultimately trained and stored in a .h5 format for potential later use.

E. Model Training and Validation

The fit function is used by inputting the train image, validating the image, setting the era size, and the steps of the era. Finally, the performance index is evaluated, and these steps are repeatedly performed by changing the parameter values using undiscovered data until the best combination of parameters is found to obtain a generalized accurate model.

III. RESULTS AND DISCUSSION

The objective of this research is to assess the effectiveness of different deep learning architectures commonly employed today, including CNN, VGG16, Mobile Net, and Inceptionv3. These models were constructed, optimized, and trained successfully. Table III presents the validation and training accuracy of each model, as well as the time taken for each model to complete training during each epoch.

TABLE III. MODEL ACCURACY

Model	Training situation	Verification situation	Mean value
	Accuracy	Accuracy	The time of each period
CNN	0.9018	0.8968	3652s
VGG16	0.8923	0.7894	6341s
Mobile Net	0.9112	0.9078	2949s
Inceptionv3	0.8734	0.8628	6068s

A confusion matrix was created for each of the deep learning models, as shown in Fig. 6, 7(a), and 7(b). The matrices represent the summary of predictions made by each model. The X-axis displays the predicted labels while the Y-axis displays the true labels. The dark labels in the matrix indicate the number of correct classifications made by the model. The test data was used to create the matrix, and it shows the prediction accuracy of each category. As is exemplified in Fig. 4, the bottom row of the confusion matrix indicates that all specimens with yellow leaf curl disease had been correctly categorized, consequently yielding a 100% predictive accuracy. The performance of the predictions for each disease category can be evaluated using these matrices. Despite being limited by space, the confusion matrices for Inceptionv3 and other CNN models can be viewed in Fig. 5(a) and 5(b).

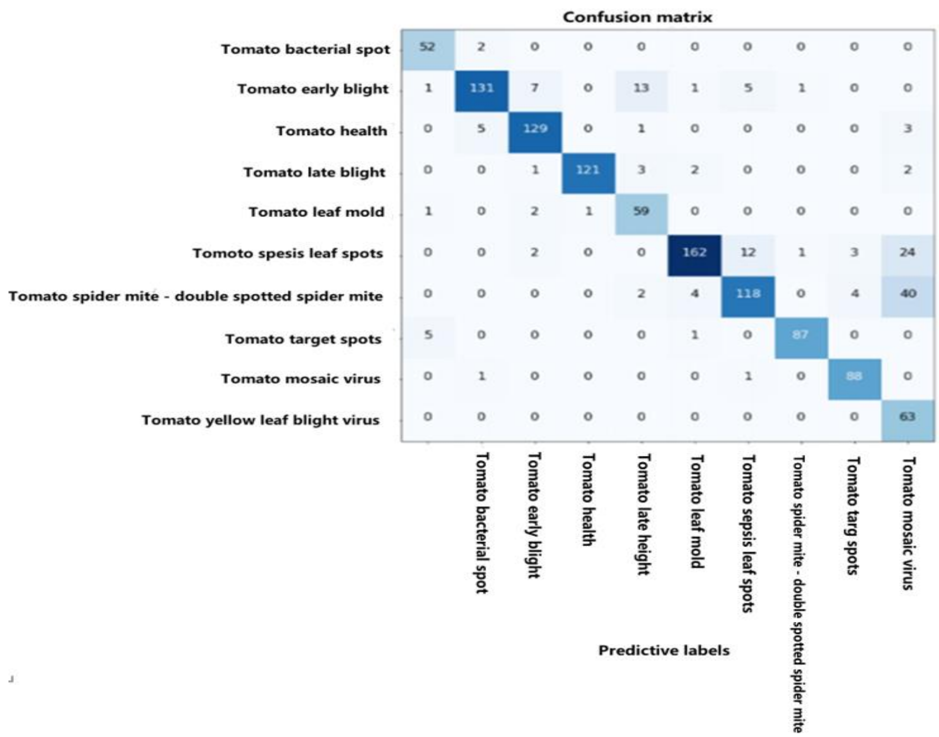

Fig. 4. Confusion matrix of MobileNet.

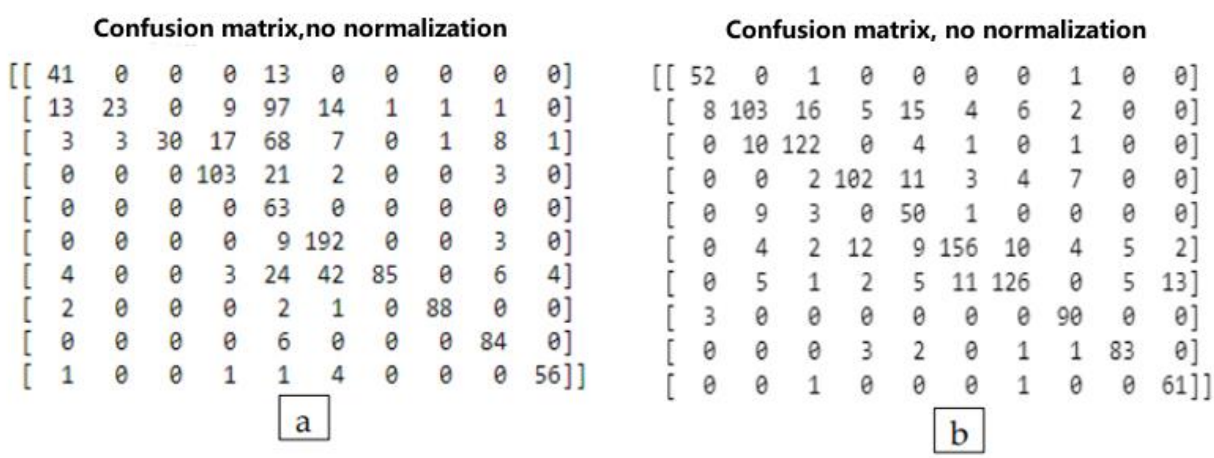

Fig. 5. Confusion matrix of (a) Inceptionv3 and (b) CNN.

The Matplotlib drawing library in Python was utilized for the creation of Fig. 6 and 7. Fig. 6 and 7 illustrate the progression of training and validation accuracy/loss of each model over time.

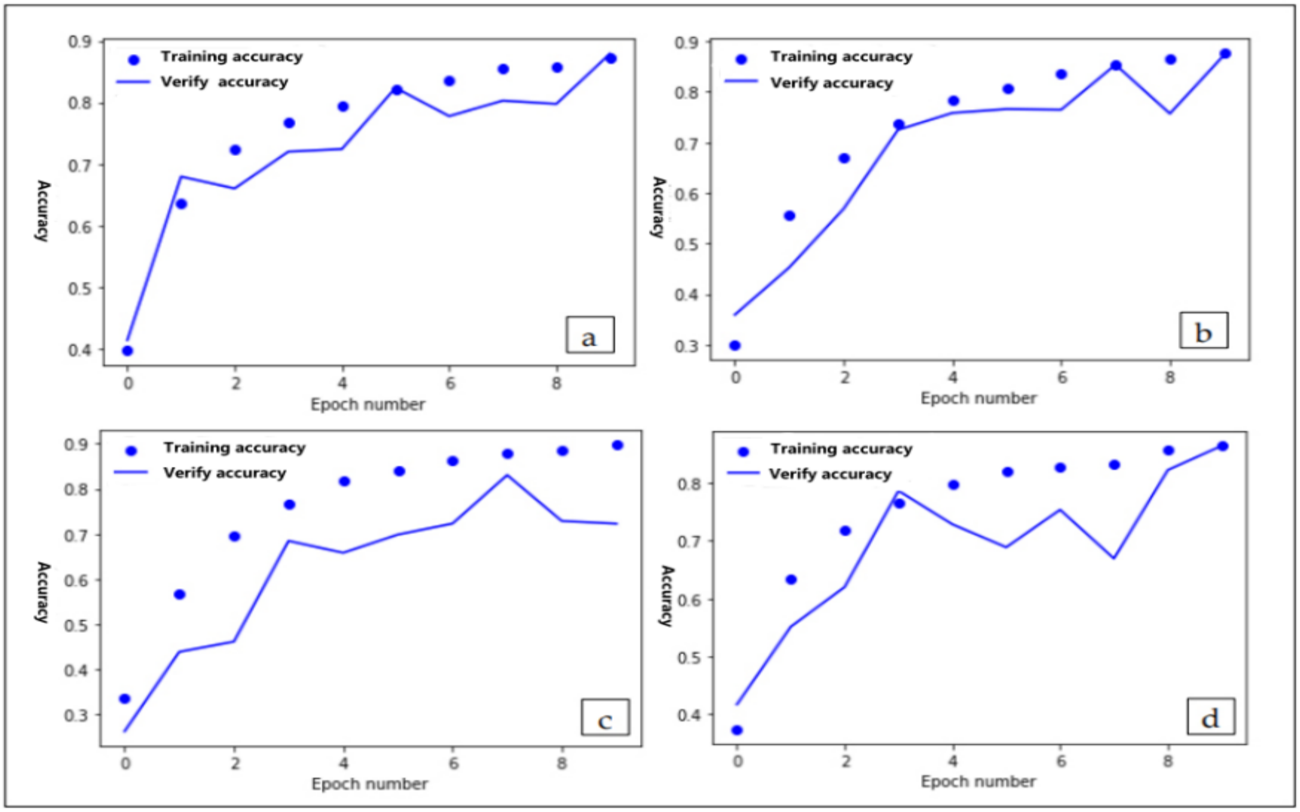

Fig. 6. Training and verifying the accuracy of the model: (a) CNN, (b) MobileNet, (c) Vgg16 and (d) Inceptionv3.

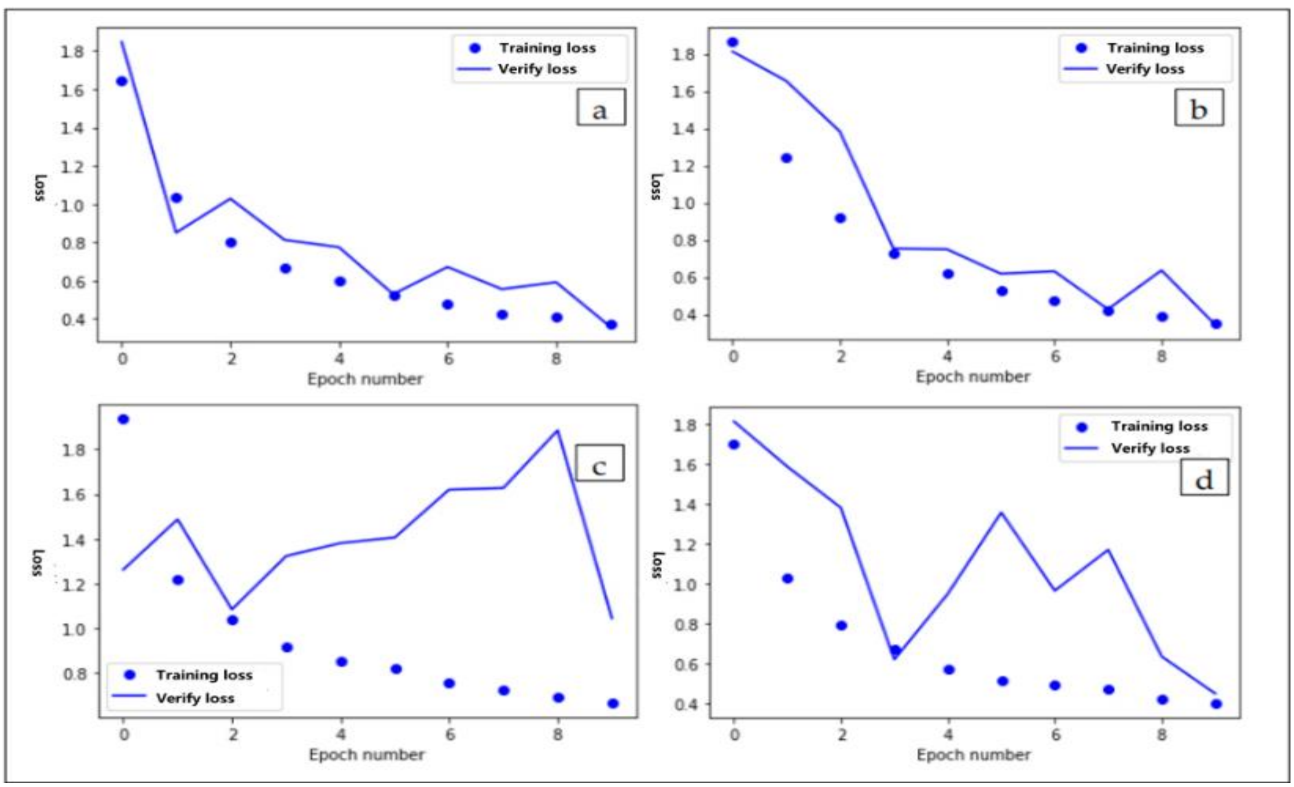

Fig. 7. Loss of training and validation models: (a) CNN, (b) MobileNet, (c) VGG16 and (d) Inceptionv3.

Referring to the diagrams, we can observe the fluctuations in verification accuracy and loss for each training epoch. To evaluate the performance of the final trained model, an unseen image is fed to the model, and the predicted category and confidence are obtained. This is achieved through the use of OpenCV, a Python library. The purpose of this is to verify if the final model can accurately predict the unseen image. In Fig. 8(a), an image of a leaf affected by early blight is shown as an example.

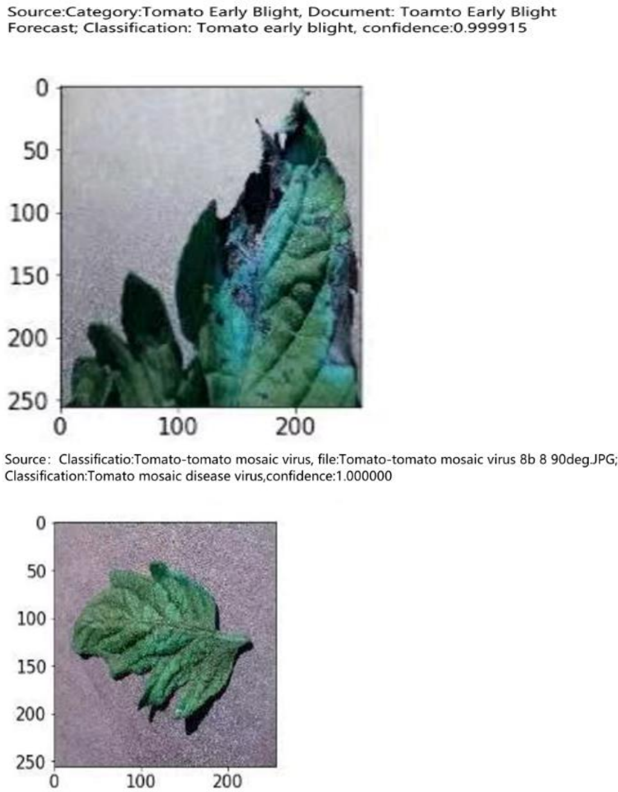

Fig. 8. Prediction of the MobileNet model for (a) Mosaic virus and (b) early blight.

It was correctly classified into the correct category with a confidence level of one hundred percent. Fig. 8(b) depicts a leaf infected by mosaic virus. In addition, it was confidently classified into the correct category.

A summary of correctly and incorrectly classified images is presented in Table IV. The initial column of the table denotes the various disease categories, while the second column illustrates the sum of images presented to the Mobile Net model from each category. The third column signifies the sum of images accurately classified, while the fourth column indicates the aggregate of images misclassified.

Table IV presents a synthesis of the correct and erroneous classification of images for each disease type, coupled with the overall number of images supplied to the ultimate Mobile Net model from each category. The last column of the table gives the average accuracy of each category. Research has demonstrated that the yellow leaf curl virus has been accurately categorized with a mean accuracy of 100%. Table IV can be utilized to likewise evaluate other diseases. The deep learning model has demonstrated remarkable enhancement in the categorization of plant diseases. This study seeks to contrast the efficacy of a Convolutional Neural Network (CNN) model developed from the ground up with that of a fine-tuning pre-training model using the Plant Village dataset from Kaggle. Training the VGG16 model was time-intensive due to the large parameter count. It can be concluded that an increase in the number of parameters also increases the time required for training the model.

TABLE IV. SUMMARY OF CNN MODEL PREDICTION

Species	Total	Correct value	Error value	Accuracy (%)
	Image	Classification value	Classification value	
Fine plaque	54	52	2	96
Early blight	159	131	28	82
Healthy	138	129	9	93
Late blight	129	121	8	94
Leaf mold	64	60	4	94
Septicemic plaque	207	165	42	80
Spider mite	168	118	50	70
Target point	94	88	6	94
Mosaic virus	90	88	2	98
Yellow leaf curl virus	84	84	0	100

When examining the training and validation accuracy of VGG16, it is evident that the former significantly exceeds the latter, indicating the model's strong performance on training data but weak performance on test data. This phenomenon is known as overfitting, which frequently results from an overly complex model. With its relatively smaller number of parameters, Mobile Net is less complex than Inceptionv3 and VGG16, making it better suited for effective detection. The prediction accuracy of Mobile Net and CNN models is 0.90 and 0.89, respectively, indicating their superior suitability for test data compared to VGG16. Fig. 6, 7(a), and 7(b) show that the classification accuracy of Fig. 6 is 100% compared to other models for the last category of matrices (yellow leaf curl). Determining the layers to freeze and train for the fine-tuning of VGG16 and Inceptionv3 necessitates more time. These models are difficult to train due to their high complexity and sensitivity pattern identification. In contrast, the proposed CNN architecture is slightly shallow, making it quicker to train on the same dataset. Accuracy is dependent on parameter identification, which requires prior experience. Mobile Net outperforms Inceptionv3 in terms of size, delay, and accuracy, and could easily be implemented in mobile devices and embedded vision applications. Although the CNN model in this study is less deep than other models, it is still easily applicable due to its reduced computational complexity.

IV. Conclusions

This study suggests that a basic convolutional model with a minimum of four layers can be enough for extracting the relevant features for categorising tomato plant diseases in comparison to other tuned models. VGG16 and Inceptionv3 have relatively more parameters, but do not reflect better recognition accuracy. This requires more time to refine, train and run on the one hand, and limits its deployment and fast recognition on smaller systems on the other. The suitability of Mobile Net and CNN for this study is due to its compactness, speed, and ability to run smoothly on mobile devices. With guaranteed high accuracy recognition, MobileNet and simple CNN structures are not only easier to deploy, adjust and work with in the areas covered by this study, their lighter weight structure allows them to have further development potential.

The aim of this investigation is to deliver a conclusive answer instead of a supposition, which will support upcoming scholars in opting for the most innovative and efficient deep learning algorithms and networks for their further research in the realm of automatic plant disease classification. Currently only tomato leaf images obtained from the internet were examined in this paper, and they will be further tested on real-time images detection program in the future, employing the CNN model developed throughout the research process.

V. Funding Statement

Project support: This project is funded by the national Guilin University of Electronic Science and Technology Student Innovation and Entrepreneurship Training Program (No.202110595052).

References

[1] P. Das & P. Misra. Artificial intelligence based automated detection of plant diseases: A review. Archives of Computational Methods in Engineering, 2019, 26(6), 1463-1486.

[2] K. Kumar, N. Belwal, and A. Singh. "A Review on Detection and Diagnosis of Plant Leaf Diseases using Image Processing Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, no. 10, 2017, pp. 63-68.

[3] M. Pautasso. Emerging infectious diseases in crops: the plant health challenge. Annals of Applied Biology, 2013, 163(1), 1-8.

[4] R. Michael Davis, Michael D. Littrell, Tomato Disease Management: A Practical Guide"

[5] R. M. Davis, R. N. Raid, Integrated Management of Tomato Diseases"

[6] S. S. Chouhan, (2018). Feature extraction and classification using Deep convolutional Neural Networks. Journal of Cyber Security and Mobility, 2018, 8(2), 261-276.

[7] Retrieved 2nd December 2020, from http://www.tomatonews.com/en/background_47.html Chouhan, S. S.

[8] A. K. Rangarajan, R. Putushothaman, &A. Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Computer Science, 2018, 133(133), 1040-1047.

[9] A. Kaul, U. P. Singh, & S. Jain. Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology. IEEE Access, 2018, 6, 8852-8863.

[10] A. F. Fuentes, S. Yoon, J. Lee, and D. S. Park. High-performance deep neural network-based tomato plant diseases and pests' diagnosis system with refinement filter bank. Frontiers in Plant Science, 2018, 9: 1162.

[11] J. Singh, A. Kaur & A. Singh, An efficient technique for classification of plant leaf diseases using decision tree. International Journal of Engineering and Technology, 2018, 7(4.34), 107-110.

[12] Y. Dandawate, and R. Kokare, "An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective." International Conference on Advances in Computing, Communications and Informatics (ICACCI 2015), 2015, 794–799.

[13] K. Mohanapriya, M. Balasubramani, "Recognition of Unhealthy Plant Leaves Using Naive Bayes Classifier." IOP Conference Series: Materials Science and Engineering, 2019, 561(1).

[14] S. N. Khan Meera, A. A. Shaikh, H. Ansari, N. Ansari, Disorder Detection in Tomato Plant Using Deep Learning. SSRN Electronic Journal, 2019, 2154–2160.

[15] H. Durmus, E. O. Gunes, M. Kirci, Disease detection on the leaves of the tomato plants by using deep learning, 6 th International Conference on Agro-Geoinformatics, 2017.

[16] S. Albawi, T. A. M. Mohammed, S. Alzawi, "A Data-Driven Approach To Precipitation Parameterizations Using Convolutional Encoder-Decoder Neural Networks Pablo." IEEE, 2017.

[17] M. A. Hossain, & G. Muhammad. Comparative performance analysis of convolutional neural networks for plant disease identification. Computers and Electronics in Agriculture, 2020, 171, 105275.

[18] H. S. Nagamani, & H. Sarojadevi. Tomato Leaf Disease Detection using Deep Learning Techniques, 2022, 13(1), 305-311.

[19] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection." Frontiers in Plant Science, 2016. 7:1419.

[20] A. R. Patil, & R. S. Sonawane, Brain tumor classification using CNN with improved training dataset. Journal of Ambient Intelligence and Humanized Computing, 2021, 12, 4573-4583

[21] Available at: https://wiki.tum.de /display/lfdv/Layers+of+a+Convolutiona Neural+Network Bodapati, J.D. and Veeranjaneyulu, N. (2019).

[22] Y. LeCun, Y. Bengio & G. Hinton, Deep learning. Nature, 2015, 521(7553), 436-444.

[23] A. Krizhevsky, I. Sutskever, & G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems, 2012. 1097-1105.

[24] K. Simonyan, & A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[25] A. G. Howard, M. Zhu, B. Chen, et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, 1704.04861.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, Rethinking the Inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 2818-2826.

[27] J. P. Too, M. A. U. Khan, M. A. Rana, & A. M. Khattak, Impact of deep learning algorithms on plant disease identification: A review. Computers and Electronics in Agriculture, 2019, 162, 707-723.