Applied Computer Science, vol.18, no. 2, pp. 56–70
doi: 10.35784/acs-2022-13

Submitted: 2022-05-16 | Revised: 2022-06-17 | Accepted: 2022-06-21

Keywords: Leaf Disease Detection, Convolutional Neural Network, Deep Learning, Transfer Learning.

Mahmoud BAKR[*], *Sayed ABDEL-GABER*[**], *Mona NASR*[**], *Maryam HAZMAN*[*]

TOMATO DISEASE DETECTION MODEL BASED ON DENSENET AND TRANSFER LEARNING

Abstract

Plant diseases are a foremost risk to the safety of food. They have the potential to significantly reduce agricultural products quality and quantity. In agriculture sectors, it is the most prominent challenge to recognize plant diseases. In computer vision, the Convolutional Neural Network (CNN) produces good results when solving image classification tasks. For plant disease diagnosis, many deep learning architectures have been applied. This paper introduces a transfer learning based model for detecting tomato leaf diseases. This study proposes a model of DenseNet201 as a transfer learning-based model and CNN classifier. A comparison study between four deep learning models (VGG16, Inception V3, ResNet152V2 and DenseNet201) done in order to determine the best accuracy in using transfer learning in plant disease detection. The used images dataset contains 22930 photos of tomato leaves in 10 different classes, 9 disorders and one healthy class. In our experimental, the results shows that the proposed model achieves the highest training accuracy of 99.84% and validation accuracy of 99.30%.

1. INTRODUCTION

Agriculture is a major component of the Egyptian economy, contributing up to 11.3 percent of Gross Domestic Product (GDP) and 28 percent of all jobs. The economy relies on agricultural product quality, which is affected by weather and other environmental factors. Since a wide range of agricultural products are produced and exported to many countries, it is vital to generate high-quality products with an acceptable yield. Over 80% of the human diet is comprised of plants production. Plants are afflicted by a variety of plant diseases, including bacteria, fungus, and viruses. According to Food and Agriculture Organization (FAO), plant pests and diseases are responsible for losses of 20 to 40% of global food production (Plant Health and Food Security, 2017). For Egypt, assisting in the resolution of this issue is a huge challenge helping to achieve food security.

[*] Climate Change Information Center and Expert Systems, Agricultural Research Center, Egypt, mah.bakr.2005@gmail.com
[**] Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt, m.nasr@helwan.edu.eg

In the case of farming, disease mitigation has recently become a significant factor. Plant disease identification is essential in the field of agriculture since plant diseases are unavoidable. The most diseases' symptoms are appeared on plant leaves. So, checking the status of a diseased plant's leaves is the simplest technique to figure out if it's infected. Plant disease recognition is a challenging task for agriculture specialists to tackle since it necessitates the use of scientific procedures and a long period of observation (Mohamed, Abdel-Gaber, Nasr & Hazman, 2020).

The shortage of agricultural extension workers involved in providing agricultural advice and guidance to farmers has become a major problem in Egypt. So that, farmers become dependent on themselves or the Internet to solve any problem they face in agricultural operations.

As the widespread use of smart phones among farmers, as well as the widespread use of graphics processing units (GPU) in computers and servers, and the rapid advancement of artificial intelligence, computer vision, and deep learning techniques, it becomes a necessary to develop an automated system that can perform plant disease recognition operations and provide an effective solution.

Plant diseases can be detected using CNNs (Venkatesh et al., 2020). CNN is one of the most powerful pattern identification techniques for massive data sets. CNN has a really encouraging performance in terms of detecting these disorders. Various CNN classification architectures, VGG16, Inception V3 and DenseNet201 were previously used in diseases detection (Venkatesh et al., 2020; Peyal et al., 2021).

In order to develop an automatic plant leaf disease detection, a comparison study is applied to find the high accuracy CNN deep learning models. Then the one with highest results is used as the base transfer learning model for our proposed model. The base transfer learning model works as feature extraction followed by a CNN classifier. The contributions of this research are:

a) comparison among some transfer learning based models,
b) proposed model based on DenseNet transfer learning as features extractor and a CNN classifier.

In the presented study, Tomato leaves images a subset of the plantvillage images dataset are used. It includes 22930 photos for 10 different classes downloaded from kaggle website (Kaggle, 2018). These classes are: tomato Bacterial spot, tomato early blight, tomato late blight, tomato leaf mold, tomato septoria leaf spot, tomato spider mites two spotted spider mite, tomato target spot, tomato yellow leaf curl virus, tomato mosaic virus and tomato healthy. First, the selected tomato images dataset is resized and augmented to be ready for training the classification model. In order to improve the classification results, we do some fine-tuning to classification models and rerun the classification models. Then, we test different models against subset of images. The DenseNet model gives the highest accuracy used as features extractor to our proposed CNN model. Finally, we compared the results and analyze them.

The paper is arranged as follows: Section 2 describe previous related works. Methodology has been explained in Section 3. Section 4 highlights the proposed model. Section 5 holds experimental result and analysis. Concluding remarks is in Section 6.

2. LITERATURE REVIEW

Over the years, there has been debate on how to detect plant diseases. Many researchers have used machine learning approaches to construct a variety of acceptable designs for detecting plant diseases.

The authors in (Rangarajan, Purushothaman & Ramesh, 2018) utilised AlexNet and VGG16 to classify six different tomato diseases as well as a healthy class. The performance was assessed by changing the number of images, batch sizes, and weight and bias learning rates. They concluded that AlexNet outperforms VGG16 in terms of accuracy and execution time. It should be noted that, given that this work is also aimed at the classification of diseases found in tomato plants. Their proposed methodology was developed based on the results reported by this comparison, allowing support in the delimitation of the work and selection of architectures to implement. While being able to discard the implementation of VGG16 due to the disadvantages it presents in comparison to AlexNet, particularly in the computable domain.

The authors in (Hong, Lin & Huang, 2020) used transfer learning to reduce the size of the training data, the time and the computational costs when building deep learning. They classify 9 types of disease leaves including healthy tomato leaves. Five deep network structures of Resnet50, Xception, MobileNet, ShuffleNet and Densenet121_Xception were applied to perform the feature extraction. Those network structures with different learning rates were compared in experiment. Adjust the appropriate training parameters and test those networks. Compared the five convolutional neural network, the parameters and the average accuracy are different. The best recognition accuracy of Densenet_Xception is 97.10%, but the parameters of Densenet_Xception are at most. The recognition accuracy of ShuffleNet is 83.68%, and the paramenters are small.

The authors in (Kabir, Ohi & Mridha, 2020) investigated an optimal plant disease identification model combining the diagnosis of multiple plants. They used data that collected from various online sources and it included leaf images of six plants: tomato, potato, rice, corn, grape, and apple. They implemented numerous popular convolutional neural network (CNN) architectures. They found that the Xception as well as DenseNet architectures perform better in multi-label plant disease classification tasks.

The authors in (Agarwal et al., 2020) applied a CNN based approach for the disease detection and classification of Tomato. The experimental results shows the efficacy of the proposed model over pre-trained model i.e. VGG16, InceptionV3 and MobileNet. The classification accuracy varies from 76% to 100% with respect to classes and average accuracy of the proposed model is 91.2% for the 9 disease and 1 healthy class.

The authors in (Afifi, Alhumam & Abdelwahab, 2021) developed and evaluated several methods for identifying plant diseases with little data. They used three CNN architectures (ResNet18, ResNet34, and ResNet50) to build two baseline models, a Triplet network and a deep adversarial Metric Learning (DAML) approach. These approaches were trained from a large source domain dataset and then tuned to identify new diseases from few images, ranging from 5 to 50 images per disease. Their proposed approaches were evaluated in the case of identifying the disease and plant species together or only if the disease was identified, regardless of the affected plant. The results show that the baseline model achieved an accuracy of 99% when the shift from source domain to target domain was small and 81% when that shift was large and outperformed all other competitive approaches.

The authors in (Ji, Zhang & Wu, 2020) proposed a CNN model to identify grape diseases from images into 4 classes. The proposed model is a united CNNs architecture based on Google InceptionV3 and ResNet50 called UnitedModel. UnitedModel takes advantage of the combination of InceptionV3's width and ResNet50's depth and learn from the output features layers from both models. The proposed UnitedModel achieves 99.17% accuracy.

In this paper, we compare four different deep learning models based on transfer learning. As our pre-trained model in Transfer Learning, we mostly employed the DenseNet201 network, a common CNN architecture. Several Transfer Learning architectures were also examined with a few additional well-known pre-trained models (VGG16, Inception V3 and ResNet152V2) and compared to DenseNet201. Additionally, Fine-Tuning has been performed to improve the detection accuracy. The dataset in our experiment includes 9 different diseases as well as the images from healthy plants. Our method for detecting plant diseases is presented in the below section.

3. METHODOLOGY

In this paper, several types of supervised deep learning techniques are utilized to detect tomato leaves diseases. We aim to explore their performance in detecting the 10 considered tomato diseases and concluding the best of them. Then we will use the best model as the base model for our proposed model.

Several steps are necessary for the implementation of deep learning models. The data set is first collected, then divided into two portions, usually 80 percent training and 20 percent validation. Deep learning models are then trained from scratch or using the transfer learning technique, and training/validation plots are created to determine the models' significance. The images are next classified using performance metrics (type of plant disease), and finally, visualization techniques/mappings are utilized to classify the images (Saleem, Potgieter & Arif, 2019).

It has been demonstrated that CNNs do not require pre-processing, feature extraction, or feature classification in order to perform image recognition. The trained model, on the other hand, can swiftly classify the image. The training of a large-scale neural network takes a long time, and it requires a massive number of data sets. Also manually labelling data according to specified selection criteria is laborious and expensive (Chen et al., 2020).

When developing deep learning models, transfer learning is a knowledge sharing strategy that decreases the size of the training data, the time, and the computing cost. Transfer learning allows a pre-trained model's learning to be transferred to a new model. Transfer learning is a machine learning approach in which CNNs trained for a task is reused as the starting point for a model on another task (Peyal et al., 2021; Chen et al., 2020).

In order to compare between different models, we follow the steps shown in Fig. 1. First, we select the tomato images dataset, which is a subset form plantvillage images dataset. Then the required preprocessing of images like resizing is applied. After augmenting the images, we build the training model using the selected classification models. To improve results of the used models, we do some fine-tuning to the classification models and retrain them to get better results. Then we test the selected models against subset of images. Finally, we analyse the findings and the results.

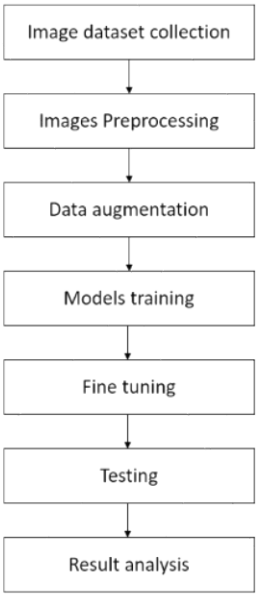

Fig. 1. Steps of Detection and classification process for leaf diseases

3.1. Tomato leaves images dataset

Tomato Dataset is a subset of the larger plantvillage dataset (Kaggle, 2018). It contains 22930 images, divided into three sets: 75% for training, 20% for validation and 5% for testing. A tomato leaf appears in every image in the data set, and the leaf takes up the majority of the image's space and provides an almost constant background. The data set divided into 10 classes, 9 classes of tomato diseases beside the tomato healthy class. The 10 classes were as follow: tomato Bacterial spot, tomato early blight, tomato late blight, tomato leaf mold, tomato septoria leaf spot, tomato spider mites two spotted spider mite, tomato target spot, tomato yellow leaf curl virus, tomato mosaic virus and tomato healthy. They are shown in Fig. 2. Table 1 shows the number of images for each disease.

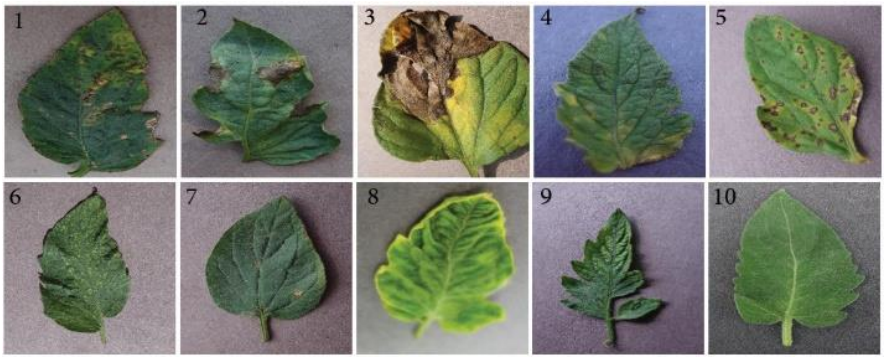

Fig. 2. Examples of tomato 10 classes – (1) Bacterial spot, (2) Early blight, (3) Late blight, (4) Leaf mold, (5) Septoria leaf spot, (6) Spider mites two spotted, (7) Target spot, (8) Yellow leaf curl virus, (9) Mosaic virus and (10) Tomato healthy

Tab. 1. Images count for each class of Tomato dataset

Tomato Class	Training Images	Validating Images	Testing Images
Bacterial spot	1,617	425	85
Early blight	1,824	480	96
Late blight	1,758	463	93
Leaf mold	1,788	470	94
Septoria leaf spot	1,658	436	87
Spider mites Two-spotted	1,654	435	87
Target spot	1,736	457	91
Yellow leaf curl virus	1,863	490	98
Mosaic virus	1,700	448	90
healthy	1,830	481	96

3.2. Image preprocessing

Image preprocessing enhances the quality of the image data needed for image classification. Geometric transformations of images, such as image rotation, scaling, and translation, are used in preprocessing approaches. In this step, we decreased the resolution of all of the images to 224*224 pixels during the preprocessing stages, the original images are 256*256 pixels. It must ensure that all images are of the same size and resolution.

3.3. Augmentation Process

CNN requires a large amount of training data to achieve improved results (Shorten & Khoshgoftaar, 2019). In order to improve the model's performance, image augmentation is frequently required to create the best deep CNN model with insufficient training data. Image augmentation increases the amount of images in the data set and reduces overfitting by adding a few distorted photos to the training data. When the network learns the data rather than the overall pattern of the dataset, this is known as overfitting. Image augmentation artificially creates training images using a range of processing methods or a combination of processing methods such as image flipping, rotation, blur, relighting, and random cropping (Chen et al., 2020). In our study we do the following for images augmentation: scaling the images, shearing, zooming and horizontal flipping.

3.4. Fine-tuning

Fine-tuning is a technique for improving a function's efficiency. It makes little adjustments to improve the outcome. The adjustment process is so important that even minor changes have a significant impact on the training process in terms of computation time, convergence speed, and the number of processing units used (Too, Yujian, Njuki & Yingchun, 2019). This fine-tuning process was repeated several times to improve the accuracy of our model. The parameters used for training and fine-tuning that give best results are as shown in Table 2.

Tab. 2. Fine-Tuning parameters and values used through training models

Parameter	Value
Batch size	32
Steps per epoch	545
Epoch	50
Validation steps	1
Optimizer	Adam
Activation function	Softmax

3.5. Training the models

In this step, the selected CNN models trained on the data set of tomato diseases identification. During the training process, the fixed low-level network parameters are unchanged, the high-level network parameters are fine-tuned. The tomato disease image is input into the network to train the high-level parameters of the network, and the trained model is used to classify the 10 classes of tomato leaves. The selected CNN are VGG16, Inception V3, Resnet152V2 and DenseNet201.

3.5.1. VGG16

The VGG architecture was introduced in 2014 by Simonyan and Zisserman of Oxford University's Visual Geometry Group and Google DeepMind. It's popular because it's straightforward, with only 16 convolutional layers stacked on top of one another. It features two fully connected layers with 4096 nodes each and a softmax classifier, as well as max-pooling layers that help reduce volume size (Simonyan & Zisserman, 2015). VGG16 is made up of thirteen convolution layers, including five combined max-pooling layers and three completely connected layers, according to their research. The rectified linear unit (ReLU) function comes after the second fully connected dense layer. The network's last layer is a softmax regression classifier, which uses probability to classify the input images. For VGG16 architecture, the image input size is appointed to 224 x 224 x 3. Fig. 3 shows the architecture of VGG16 model. In our study, for transfer learning of VGG16, the last layer with 1000 output classes was deleted and the model output was flattened then, a dense layer with 10 outputs -tomato classes- was added to the model.

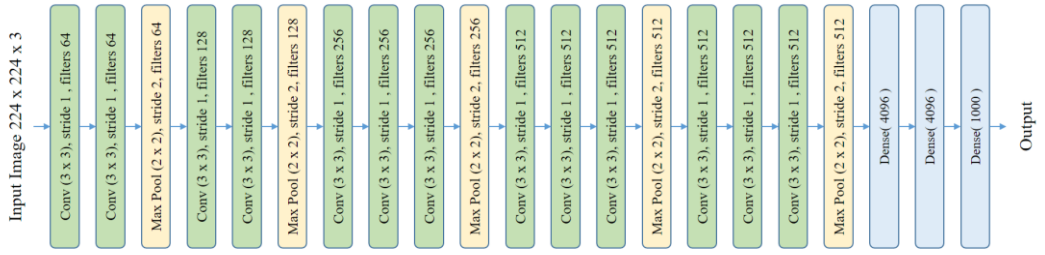

Fig. 3. VGG16 architecture

3.5.2. Inception V3

The deep convolutional architecture Inception V3 is commonly utilized for classification problems. Szegedy and his colleagues presented their model concept based on the GoogleNet design (Szegedy et al., 2016). By changing the inception module, Inception V3 was created. Each block of the Inception V3 network comprises several symmetric and asymmetric building blocks, as well as various branches of convolutions, average pooling, max pooling, concatenated, dropouts, and fully-connected layers. Because there are 42 layers and 29.3 million parameters in this network, the computational cost is just 2.5 times that of GoogleNet. Finally, the scientists discovered that by reducing the number of parameters and further regularising the network with batch normalised auxiliary classifiers label smoothing, they can train a high-quality network on tiny training sets (Szegedy et al., 2016). Fig. 4 shows the architecture of Inception V3 model. In our study, for transfer learning of Inception V3, the top layer was deleted and the model output was flattened then, a dense layer with 10 outputs -tomato classes- was added on the top of the model.

Fig. 4. Inception V3 architecture

3.5.3. Resnet152V2

A CNN architecture with hundreds or thousands of convolutional layers is known as a Residual Network (ResNet) (Gulli & Pal, 2017). Additional layers' efficacy was reduced by previous CNN configurations. ResNet has a large number of layers and is extremely fast. The main difference between ResNetV2 and the original (V1) is that V2 applies batch normalization to each weight layer before applying it. ResNet has great performance in image recognition and localization tasks, demonstrating the importance of numerous visual recognition tasks (Kumar, Arora, Harsh & Sisodia, 2020). Fig. 5 shows the architecture of Resnet152V2 model which include 152 layers in depth and build mainly from 3-layer blocks. In our study, for transfer learning of Resnet152V2, the last output layer was deleted and the model output was flattened then, a dense layer with 10 outputs – tomato classes – was added to the top of the model.

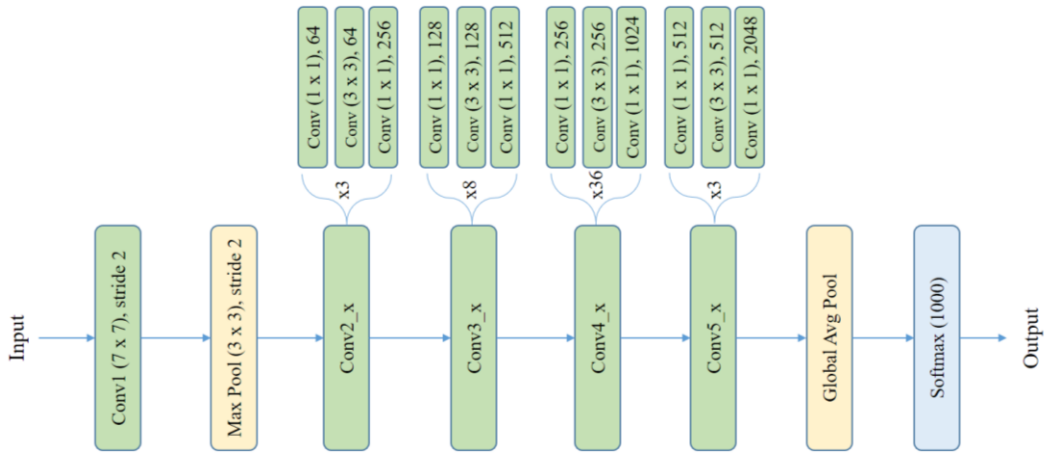

Fig. 5. ResNet152V2 architecture

3.5.4. DenseNet201

Authors in (Huang, Liu & Weinberger, 2016) proposed a highly linked convolutional network design in their study. All layers in the network are connected directly to each other in a feed-forward manner to enable maximum information flow between them. All previous layers' feature-maps are utilized as inputs into each layer, and its own feature-maps are used as inputs into all subsequent layers. DenseNets solves the vanishing-gradient problem while drastically reducing the number of parameters. Fig. 7 shows the architecture of DenseNet201 model which explained in more detailed in the following section.

4. PROPOSED MODEL

In this section, the proposed model based on a pre-trained model and CNN classifier is designed for the prediction and classification of tomato diseases from infected leaves images. The pre-trained architecture utilized in the proposed model is DenseNet201 as it gives the highest accuracy among other models. DenseNet201 is used to extract features, which are then fed into a CNN for classification. The test set and validation set are then used to evaluate the proposed model. As shown in Fig. 6, the suggested model contains five phases. The first phase is data pre-processing. The second phase is data augmentation. The feature extraction phase, which uses the pre-trained architecture DenseNet201 with transfer learning, is the third phase. The classification of tomato leaf diseases using the retrieved features and the CNN classifier is the fourth phase. The final phase is performance measurement and analysis. The first two phases of images pre-processing and data augmentation are done the same way as explained in section 3.

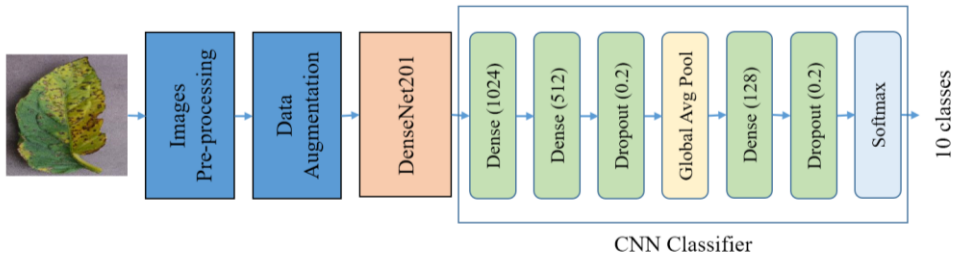

Fig. 6. The proposed model

In the third phase, the DenseNet201 model was proposed, which uses transfer learning to extract features automatically and leverage their weights learnt on the ImageNet dataset to reduce calculation workload. DenseNet201's architecture allows for the creation of simple and straightforward models. It's also feasible to reuse features across layers, making the architecture's parameters more efficient and allowing for more variation in subsequent layers and improved performance. The architecture connects each layer to all other layers in a feed-forward approach. Additionally, the DenseNet201 model uses a pooling layer and bottleneck structure. As a result, this architecture minimizes model complexity and property parameters, making it more efficient. Each layer of DenseNet201 network implements a nonlinear transformation, and the nonlinear transformation includes the convolution (Conv), pooling, rectified linear units (ReLU) and batch normalization (BN) (Huang, Liu & Weinberger, 2016). Unlike other networks, the output of each layer is used as the input for each subsequent layer in the Densenet201 network (i.e., X0, X1, X2, X3 and X4), so there are L(L+1)/2 connections in an L-layer DenseNet201 network (Huang, Liu & Weinberger, 2016). In the current study, the DenseNet201 architecture contains 707 layers and about 20 million parameters. The numbers of parameters of different models used in this study are shown in Table 3. The images dimensions in the input layer are set to 224 x 224 x 3. Fig.7 shows the architecture of DenseNet201.

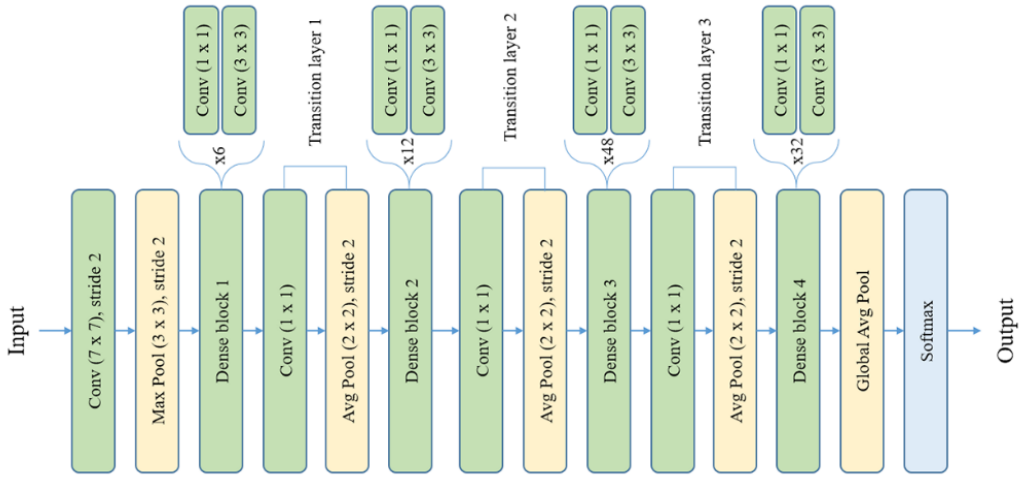

Fig. 7. DenseNet201 architecture

In the fourth phase, the classification output layers of the DenseNet201 network are removed in the fourth phase, and six layers for the classification task are proposed. Fig. 6 depicts the architecture of the suggested model based on DenseNet201. The first layer is a dense layer with 1024 neuron and a Rectified Linear Unit (ReLu) as an activation function. The second layer is also a dense layer with 512 neurons and activation ReLu. To avoid overfitting, the third layer is a dropout layer with a dropout rate of 0.2, which indicates that 20% of the neurons will output 0. The fourth layer is a global average pooling layer for size reduction of features maps. In the fifth layer is a dense layer with 128 neurons and activation ReLu. The sixth layer is a dropout layer with dropout rate 0.2. The last layer is a dense layer with 10 neurons and Softmax activation function. The last layer output the 10 classes of tomato diseases. In the following section, we go through the results of our proposed model in detail and compare it to other transfer learning-based models.

5. EXPERIMENTAL AND DISCUSSIONS

Our experiment is implemented using Jupyter Notebook (Jupyter.Org, 2021) which is an open-source web application. It has several algorithms' coding for both feature extraction and classification. Also, it includes code for data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. The machine over which this research has been accomplished is having an NVIDIA GeForce RTX 2060 graphic card with dedicated 6.0 GB of RAM and 1920 CUDA Cores. Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz. Memory: 16.0 GB.

In our experiment first: tomato leaves images from plantvillage dataset were resized into 224×224. Then the augmentation was performed. We used the weights of imagenet for saving time of training and getting higher accuracies (Huang, Liu & Weinberger, 2016). We used Adam optimizer, softmax activation function and batch size equals 32. Learning rate and other parameters was set to default values.

Then, we used four CNN models, VGG16, Inception V3, ResNet152V2 and DenseNet201, with transfer learning technique to classify tomato diseases and compare them with our proposed model. Fig. 8 shows the accuracy and loss of different models.

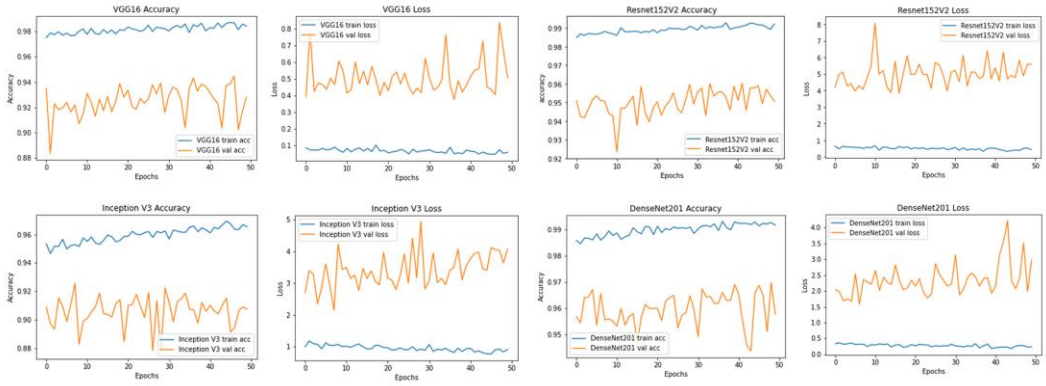

Fig. 8. Accuracy and loss of different models

Tab. 3. Results of different models

Model	Parameters (M)	Training Accuracy	Validation Accuracy	Loss	Val. Loss	Time/Step (Seconds)
VGG16	15	0.9869	0.9447	0.0466	0.4049	360.661
Inception V3	22.3	0.9560	0.9258	0.9437	2.1475	146.269
Resnet152V2	59.34	0.9909	0.9603	0.4428	4.5758	225.413
DenseNet201	19.30	0.9910	0.9698	0.3089	1.9804	164.300
Proposed Model	20.88	0.9932	0.9797	0.0230	0.0898	173.327

The results of these experiments are shown in Table 3. It shows that our proposed model has high accuracy. It achieved the highest training accuracy of 99.32 % and validation accuracy of 97.97 %.

The obtained results of the five models are analyzed to enhance the results. However, the architectures evaluation is made depending upon two parameters, validation accuracy and confusion matrix. The term validation accuracy defines how accurately the trained model is tracking the trained data. On the other hand, the sum of each column within a confusion matrix corresponds to the false positive rate (FP), and the false-negative rate (FN) for each class corresponds to each row's amount. The diagonal numbers represent the exact positive rate (TP), and the exact negative rate (TN) is the sum of all other diagonal numbers. Nevertheless, the formulas applied to measure accuracy, precision, recall, and F1 score of an architecture are as follows:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{1}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{2}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{3}$$

$$F1\_Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4}$$

The classification accuracy is a standard performance measure used to evaluate the efficacy of the classifier. Where TP (true positive)-correctly classified positive samples, TN (true negative)-correctly classified negative samples, FP (false positive)-misclassified negative samples and FN (false negative)-misclassified positive samples.

Tab. 4. Accuracy, Precision, Recall and F1 score for different models

Model	Accuracy	Precision	Recall	F1 Score
VGG16	0.9447	0.94	0.93	0.93
Inception V3	0.9258	0.93	0.92	0.92
Resnet152V2	0.9603	0.95	0.95	0.95
DenseNet201	0.9698	0.96	0.96	0.96
Proposed Model	0.9797	0.97	0.97	0.97

To evaluate the efficiency of different models, performance parameters such as Classification Accuracy, Precision, Recall, F1-Score are calculated as shown in Table 4. Fig. 9 shows the accuracy and loss of the proposed model. Fig. 10 shows the confusion matrix for the accuracy of the performance of the test data for the classified 10 plant leaf diseases by using proposed model. The results of the test data set is nearly the same as the results of validation data set.

Since the DenseNet201 model achieved the highest accuracy in transfer learning classification among other models and other properties of DenseNet mentioned in section 4, we used it as the base transfer learning model for features extraction to build our proposed model for tomato disease identification. DenseNet201 model also has small number of training parameters in comparison with the Resnet152V2 model as shown in Table 3 which is giving also better accuracy and that affecting the size of the model and the training time.

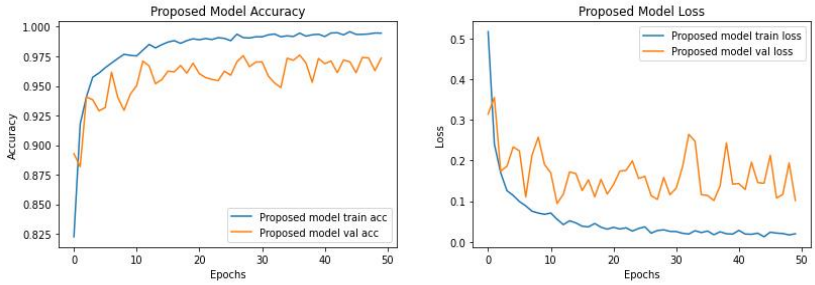

Fig. 9. Accuracy and loss of Proposed Model

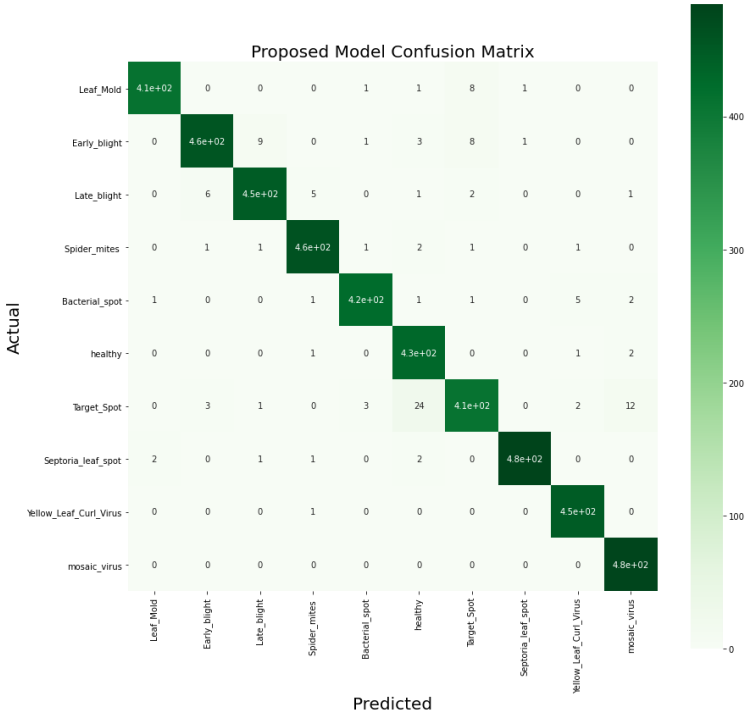

Fig. 10. Confusion matrix for proposed model

In our case, DenseNet201 contains 707 layers. One of the benefits of DenseNet201 model architecture is that we can train all layers or some layers or only the last level of layers. As we did in the first phase and get the results shown in Table 3. We did some fine-tuning by retraining about half layers of the DenseNet201 model layers – 300 layers – in the phase of features extraction and get better training accuracy of 99.84%, validation accuracy of 99.30% and testing accuracy of 99%, Training time per step: 190.349 second, validation loss: 0.0866. Table 5 shows the results of proposed model after retraining some layers.

Tab. 5. Proposed model performance values after training some layers of base model

Performance Metrics	Value
Training Accuracy	99.84%
Validation Accuracy	99.30%
Validation Loss	0.0866
Training time per step	190.349 seconds
Testing Accuracy	99.0%
Precision	0.99
Recall	0.99
F1 Score	0.99

6. CONCLUSION

In this paper, a comparison study has been conducted to find the best deep CNN model for using in plant leaves diseases detection. Four deep CNN models, DenseNet201, VGG16, Inception V3 and ResNet152V2 were trained and tested using the tomato leaf disease data set. Applying transfer learning technique to save time and effort in training these models. The data set was split into 75% for training, 20% for validation and 5% for testing, and were labelled with 10 different classes of diseased including healthy tomato leaves images. The results for each case are presented in Table 3. We also proposed a classification model based on DenseNet201 and transfer learning. In our proposed model the DenseNet201 model works as features extraction phase that followed by a CNN classifier. The results shows that the proposed model gives the highest accuracy. We apply additional fine-tuning by training some additional layers of the model during transfer learning not only the last level of layers. Finally, the results show that: the parameters and the average accuracy of the five convolutional neural networks are different. Our proposed model gives the highest accuracy.

In future work, we plan to expand our research with other pre-trained CNNs to solve multi-classification tasks. Apply our proposed model to more plants and diseases. Build a plant disease diagnosis application that help the farmers in Egypt.

REFERENCES

Afifi, A., Alhumam, A., & Abdelwahab, A. (2021). Convolutional Neural Network for Automatic Identification of Plant Diseases with Limited Data. *Plants*, *10*(1), 28. https://doi.org/10.3390/plants10010028

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, *167*, 293–301. https://doi.org/10.1016/j.procs.2020.03.225

Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, *173*, 105393. https://doi.org/10.1016/j.compag.2020.105393

Gulli, A., & Pal, S. (2017). *Deep Learning with Keras*. Packt.

Hong, H., Lin, J., & Huang, F. (2020). Tomato Disease Detection and Classification by Deep Learning. In *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* (pp. 25–29). IEEE. https://doi.org/10.1109/ICBAIE49996.2020.00012

Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. *CoRR*, *abs/1608.06993*. http://arxiv.org/abs/1608.06993

Ji, M., Zhang, L., & Wu, Q. (2020). Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Information Processing in Agriculture*, *7*(3), 418–426. https://doi.org/10.1016/j.inpa.2019.10.003

Jupyter.org. (2021). https://jupyter.org

Kabir, M. M., Ohi, A. Q., & Mridha, M. F. (2020). A Multi-Plant Disease Diagnosis Method using Convolutional Neural Network. *CoRR*, *abs/2011.05151*. https://arxiv.org/abs/2011.05151

Kaggle. (2018). https://www.kaggle.com/noulam/tomato/download

Kumar, V., Arora, H., Harsh, & Sisodia, J. (2020). ResNet-based approach for Detection and Classification of Plant Leaf Diseases. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 495–502). IEEE. https://doi.org/10.1109/ICESC48915.2020.9155585

Mohamed, A., Abdel-Gaber, S., Nasr, M., & Hazman, M. (2020). An Intelligent Approach to Mitigate Effects of Climate Change and Insects on Crops. *International Journal of Computer Science and Information Security (IJCSIS)*, *18*(3), 75–79.

Peyal, H. I., Shahriar, S. M., Sultana, A., Jahan, I., & Mondol, Md. H. (2021). Detection of Tomato Leaf Diseases Using Transfer Learning Architectures: A Comparative Analysis. In *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)* (pp. 1–6). IEEE. https://doi.org/10.1109/ACMI53878.2021.9528199

Plant health and food security. (2017). *FAO*. http://www.fao.org/3/a-i7829e.pdf

Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, *133*, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.070

Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant Disease Detection and Classification by Deep Learning. *Plants*, *8*(11), 468. https://doi.org/10.3390/plants8110468

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1409.1556

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2818–2826). IEEE. https://doi.org/10.1109/CVPR.2016.308

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272–279. https://doi.org/10.1016/j.compag.2018.03.032

Venkatesh, Nagaraju, Y., Sahana, T. S., Swetha, S., & Hegde, S. U. (2020). Transfer Learning based Convolutional Neural Network Model for Classification of Mango Leaves Infected by Anthracnose. In *2020 IEEE International Conference for Innovation in Technology (INOCON)* (pp. 1–7). IEEE. https://doi.org/10.1109/INOCON50539.2020.9298269