Evaluation of scratch and pre-trained convolutional neural networks for the classification of Tomato plant diseases

Mohammad Amimul Ihsan Aquil[1], Wan Hussain Wan Ishak[2]
[1]Department of Computer and Information Sciences, Universiti Teknologi Petronas, Malaysia
[2]School of Computing, Universiti Utara Malaysia, Malaysia

Article Info	ABSTRACT

Plant diseases are a major cause of destruction and death of most plants and especially trees. However, with the help of early detection, this issue can be solved and treated appropriately. A timely and accurate diagnosis is critical in maintaining the quality of crops. Recent innovations in the field of deep learning (DL), especially in convolutional neural networks (CNNs) have achieved great breakthroughs across different applications such as the classification of plant diseases. This study aims to evaluate scratch and pre-trained CNNs in the classification of tomato plant diseases by comparing some of the state-of-the-art architectures including densely connected convolutional network (Densenet) 120, residual network (ResNet) 101, ResNet 50, ReseNet 30, ResNet 18, squeezenet and Vgg.net. The comparison was then evaluated using a multiclass statistical analysis based on the F-Score, specificity, sensitivity, precision, and accuracy. The dataset used for the experiments was drawn from 9 classes of tomato diseases and a healthy class from PlantVillage. The findings show that the pretrained Densenet-120 performed excellently with 99.68% precision, 99.84% F-1 score, and 99.81% accuracy, which is higher compared to its non-trained based model showing the effectiveness of using a combination of a CNN model with fine-tuning adjustment in classifying crop diseases.

Corresponding Author:

Mohammad Amimul Ihsan Aquil
Department of Computer and Information Sciences
Universiti Teknologi Petronas
Persiaran UTP, 32610 Seri Iskandar, Perak, Malaysia
Email: mohammad_19000953@utp.edu.my

1.   INTRODUCTION

Healthy plants must remain protected from diseases. This helps maintain the quality and quantity of crops since they can be easily affected by diseases, leading to massive agricultural losses [1], [2]. This necessitates the need for early monitoring since it helps stop the disease in its early stages since it dictates the type of treatment that will prevent the disease from spreading into other parts of the plant or even to other plants [3]. Conventionally, the identification and detection of plant diseases has always been done through a simple observation exercise by experts [3]. However, observing the plant has to be done continuously with regular monitoring since an individual might arrive at an erroneous diagnosis based on his or her experience, showing the need for a technology that will help with precise recognition of plant diseases. There has been growing potential in a breakthrough, especially with the use of deep learning (DL), whose results are much better than some state-of-the-art architectures in providing solutions to problems by adopting the adaptability of computer vision techniques tan [4]. Ferentinos [5] defines DL as the use of artificial neural networks (ANN) architectures with multiple processing layers compared to using 'shallow' traditional ANN

architectures. DL outperforms these other architectures in its ability to work with raw data directly without relying on any hand-crafted features [3].

In deep learning, convolutional neural networks (CNNs) can be categorized under deep ANN and used in various applications such as image classification and object recognition and can implement significant improvements in the classification of images across multiple fields, including agriculture. The rapid development of CNNs paired with the DL breakthrough has served as the motivation behind the rise of many powerful architectures such as ResNet [6], Residual network (ResNet) 50, Inception V3 [7], GoogleNet [8], and AlexNet [9].

Deep neural networks have been used successfully across different applications and domains. A neural network maps an input to an output. For example, it can map an image of a diseased plant to a disease pair. The nodes present in a neural network are made up of mathematical functions that take numerical inputs and provide a numerical output, meaning that neural networks simply rely on a series of stacked layers of nodes to map out an input layer to an output. However, the main challenge involves coming up with a neural network whose edge weights, functions, and network structure can map out an input to an output correctly. By training the models, we tune their network parameters to help improve their mapping accuracy. Comparing trained and untrained models allows us to ascertain which one offers the best knowledge and how to increase the accuracy of CNN models in classifying plant diseases.

Even with the presence of required techniques to help with the detection of plant disease, there is also the need for a verified dataset containing images of both diseased and healthy plants. For a long time, such a dataset was not available while smaller datasets were only accessible at a fee. However, the rise of the PlantVillage project, which collects tens of thousands of photos of both diseased and healthy plants, has provided a free and openly available dataset [10]. For the purposes of this study, we will be using a convolutional neural approach classification containing 54,306 images of 26 diseases across 14 species. We will then rely on 34 possible classes from this dataset to test and measure the performance of our model's ability to predict plant disease.

This paper is organized into six sections. Section 2 will offer an overview of related works. Section 3 will be detailing the experiments done on the five state-of-the-art CNN architectures. Section 4 will provide the results, while discussion will be done in section 5. Section 6 will provide the instructions.

2.   RELATED WORKS

Previous research shows that the tomato fruit is one of the top fruits when ranked with over 40 vegetables and fruits based on its' relative contribution to human nutrition.' Its wide consumption worldwide is mainly attributed by its high nutrition value. For example, tomatoes are rich in vita-min E2, flavonoids, vitamin C, potassium, folate, beta-carotene, and lycopene. Tomatoes also have a wide range of health benefits, such as their cardio-protective and anti-carcinogenic properties. However, one key challenge facing the yield and quality of tomato include the various diseases during growth. Some of the common tomato diseases include Early blight (Alternariasolani) and late blight (Phytophthorainfestans). When a tomato plant is infected with any of these diseases, it spreads to the tomato leaves and can, in some cases, spread and cover the entire leaf blade. Traditionally, detecting the disease infecting a tomato plant was done through diagnostic methods and visual assessment. Some of the diagnostic methods include biomarker-based detection technology, fluorescence in situ hybridization, enzyme-linked immune sorbent assay (ELISA), and the polymerase chain reaction (PCR). However, these methods have proved destructive, inefficient, and time-consuming, necessitating the need for advanced methods for better tomato disease identification.

Recent years have seen a growth in interest regarding the research into agricultural plant disease classification. Since the development of a system that can be reliable when applied to a large number of classes has been quite challenging, this has forced a number of studies to rely on various CNNs in carrying out the detection and classification of plant diseases. For example, Kawasaki et al. [11] propose the use of CNN's in separating unhealthy cucumbers from healthy ones using the images of their leaves. In the article, the authors adopt a CNN architecture that is based on the Caffe framework [12], which includes local contrast normalization layers, pooling layers, and convolutional layers. This system managed to achieve 94.9% in terms of accuracy when assessed using the 4-fold cross-validation strategy.

Fujita et al. [13] propose the use of another approach to help with the classification of cucumber leaf disease using a CNN made up of local response normalization (LRN) operations, pooling, and four convolutional layers. The LRN used parameters drawn from the AlexNet architecture. This system would achieve an accuracy of 82.3% when assessed using the 4-fold cross-validation strategy. Sladojevic et al. [14] would then use CNNs to develop a leaf image classification plan disease recognition model. This model used the Caffe DL framework and managed to recognize approximately 13 types of plant diseases. The results from the experiment show that the model achieved a precision of between 91% to 98%, with the trained

model achieving an accuracy of 96.3%. another study done by Mohanty *et al.* [15] evaluates popular CNNs architectures GoogleNet and AlexNet by using the PlantVillage dataset to identify 26 diseases and 12 crop species [10]. However, for this study, the CNNs were measured through a different training-test distribution: 80–20, 60–40, 50–50, 40–60, and 20–80%.

Finally, when assessing the models' performance in predicting the correct crop disease pair, the best performing model achieved an accuracy of 99.35% and a mean F1 score of 0.9934. Another study done by Nachtigall *et al.* [16] investigates the use of CNNs in the identification and classification of damages and nutritional deficiencies in apple trees. The CNN architecture use was AlexNet. The authors, with the help of seven experts, compared the chosen CNN to a Multilayer Perceptron. The results from this comparison show that the CNN has an accuracy level of 97.3%. The human experts reported the CNN had an accuracy of 96%, while the MLP was less accurate at 77.3%. Brahimi *et al.* [3] would later introduce a learning CNN algorithm to help with the classification of tomato diseases. The authors relied on the standard architectures: GoogleNet and AlexNet. The study concluded that DL was far more impressive than other classification techniques such as support vector machine and random forest. In carrying out the comparison, the authors calculated the macro F-score, macro recall, macro precision, and accuracy. The CNN obtained an accuracy of 99.18%.

A study by DeChant *et al.* [4] utilizes a CNN architecture to help in the classification of leaf blight lesions from the images of maize plants. 70% of the photos were used in training the model, 15% used to help with validation, and the remaining 15% used in the actual test. The proposed system would achieve an accuracy of 96.7% on a set of test images not included in the training. Lu *et al.* [17] devised a novel CNN technique-based method to help with the identification of rice disease. The model used the AlexNet architecture to help with the identification of 10 common rice diseases. When assessed using a 10-fold cross-validation, the model achieved an accuracy level of 95.48%. Additionally, Brahimi *et al.* [3] took to carry out a comparison on the state of the art CNN architectures such as visual geometry group (VGG) 13 [6], SqueezeNet 1.1 [18], ResNet 34 [19], Inception V3, DenseNet 169 [20] and AlexNet in the classification of plant diseases. This comparison was made using three key strategies: one based on shallow strategy while the remaining two on transfer learning. The used dataset was divided into two parts: 80% for training and 20% for the actual evaluation. The final model attained an accuracy of 99.76%, with the conclusion that transfer learning was the most successful learning strategy. One key difference between this work and ours is that the ANNs used here are for comparison purposes. While the selection of ANN for this work was limited to the only three ANNs used, such as GoogleNet and two versions of ResNet, these three presented great results across the literature used.

Wang *et al.* [21], on the other hand, proposed the use of a DL approach to help with the estimation of the severity of the disease. Out of the proposed models, the best one was the VGG 16 trained with transfer learning. It posted an accuracy level of 90.4% when assessed using the hold-out test set. In another article, Wang *et al.* [22] combined transfer learning and CNN to devise a method to classify the images of crop diseases. The authors used a CNN with five convolutional layers and achieved a 90.84% accuracy. Additionally, the study also showed that the combination of transfer learning and CNN is more effective in classifying crop diseases. A study done by. Rangarajan *et al.* [23] documents the use of VGG 16 and AlexNet in the classification of six different diseases and a healthy class of tomato. The evaluation of the performance was done by modifying the bias learning rate, the weight, batch sizes, and the number of images. The findings of the study show that compared to VGG 16, the Alex Net architecture provided a better level of accuracy at the least time. It is also important to note that since this work focuses on classifying diseases in tomato plants, developing the proposed model was done using these comparison results, which allowed the selection of the architectures to be used and the delimitation of the work while also allowing the discarding of the VGG 16 implementation due to its advantages compared to AlexNet especially in terms of the computational costs.

Finally, a study done by Khandelwal and Raman [24] used different state-of-the-art approaches in the detection of plant diseases. The model used by the authors was able to attain an accuracy level of 99.374% when using transfer learning. Table 1 provides a summary of the studied methodologies and approaches that use DL models on classifying plant diseases. The results are based on Accuracy in addition to other metrics. It was also decided that the inclusion of the ANN: ResNet 50, ResNet 18, Inception V3, GoogleNet, and AlexNet be limited in making the comparison proposed in this study. The selection of the ANN was done based on the reported significant behavior supported by their corresponding results.

3.   MATERIALS AND METHODS

This section will detail the CNNs implemented to help with the classification of plant disease in tomato plant leaves for this study. The study will be looking to identify the most appropriate pertained CNN model using a 4-step procedure, including data acquisition, classification, training, and evaluation.

3.1. Data acquisition

The dataset used was acquired from the PlantVillage dataset. An open repository of 14 crops and 30 types of plant diseases detailed in over 54,323 images (9 hughes). For the purpose of this study, only images detailing tomato leaves were used. Figure 1 shows example of each sample class. The images were then resized into 256x256 pixels to help fit the network's input size. We then performed the model predictions and optimization on the downscaled images. The dataset housed 10 datasets: nine classes of tomato diseases in addition to a healthy class, which had a total of 18,160 images. The division of the images to training and testing was done based on the selected train-test set splits.
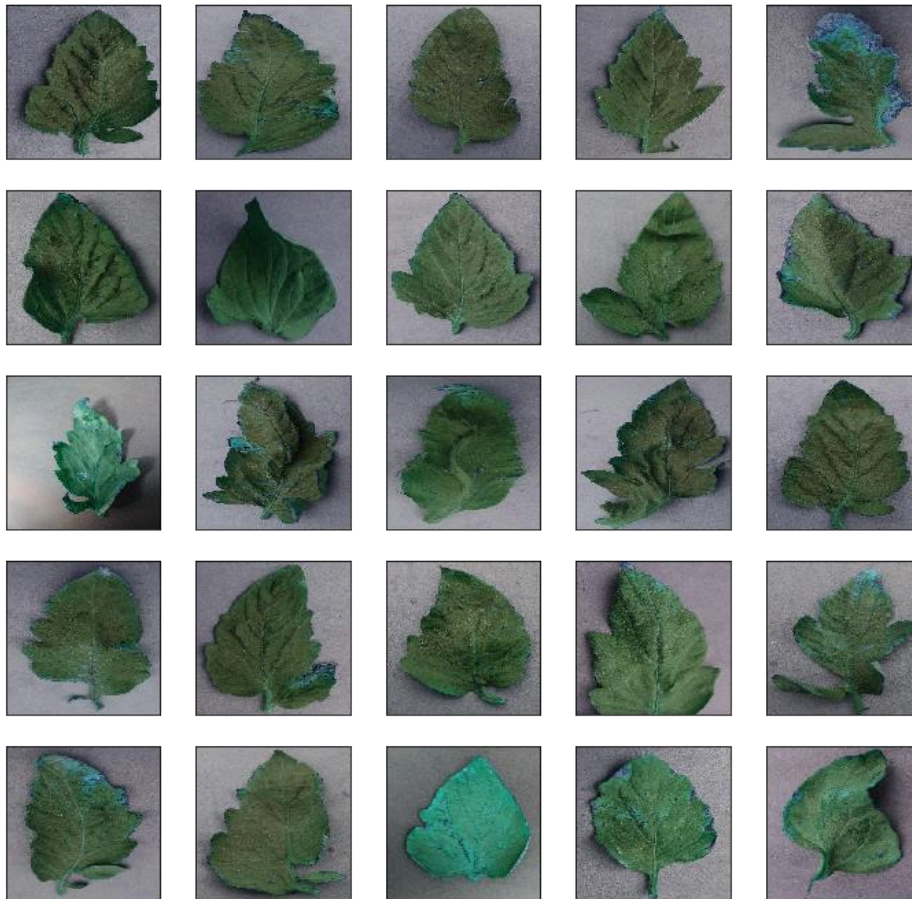

Figure 1. Sample images from PlantVillage dataset

3.2. Performance management

To gain insight into how the chosen approaches will perform on unseen data, we run the selected experiments through different train-test set splits, including 20-80 (20% of the dataset for training and 80% used for training), 40-60 (40% for training and 60% for testing), 50-50 (50% of the dataset for training and 50% for testing), 60-40 (60% for training and 40% for testing) and 80-20 (80% for training and 20% for testing). It should also be noted that the PlantVillage dataset contains multiple images of the same leaf but taken from different orientations, and such was the case for 41,112 out of the used 54,306 images. Therefore, during the test-train splits, we ensured that training was done using images of one leaf and testing done using images of a completely different leaf. Additionally, for all the experiments we did, we computed the mean F1 score, mean recall, mean precision in addition to the achieved accuracy during the training process at the end of the epoch of each regular interval. We used the final mean F-1 score in comparing the results achieved by the experimental configurations. For the purpose of analyzing the architecture's performance on the PlantVillage dataset, we started by training the models from scratch on the one hand and using transfer learning to adapt already trained models using the ImageNet dataset on the other. During the training, learning of the layers was not limited, which is always the case with transfer learning.

In simple terms, training from scratch differs from transfer learning in the initial state of weights of a few layers that allow the transfer learning approach to utilize the already available visual knowledge of the pre-trained GoogleNet and AlexNet models from ImageNet [25]. We then ran the experiments for 5 epochs (an epoch is the total training iterations whereby the neural network has managed to conduct an in-depth pass in the training set). The CNNs used in the study are VGGNet, SqueezeNet, ResNet, and DenseNet.

3.2.1. VGGNet (geometry group network)

This CNN architecture is made up of three connected layers (the FC6, FC7, and FC8layer) in addition to thirteen convolutional layers [26]. Having a combination of two 3x3 convolutional layers creates a 5x5 receptive field that has a number of kernels or learnable filters whereby each layer unit receives input from units located in the previous layer. However, this model could benefit from using more layers in increasing its expressiveness. For example, having a combination of three convolutional layers yields a 7x7 receptive field. Executing these integrated layers requires a rectified linear units (ReLU), which has an average or max-pooling operation, which is critical for multi-layer networks [27]. The representation's spatial size is decreased by pooling layers while ReLU, which in this case is the half-wave rectifier function f(x)=max(x,0), accelerates the training phase and prohibits case of overfitting [26]. The final output layer is made up of the fully-connected layers, whereby a neuron from one layer is connected to numbers in the previous volume. The fully-connected layers have a dimension of 4096. Additionally, once the while layer is once it is fed to a 1000-way softmax, it produces output probabilities for over 1000 object categories [7].

3.2.2. SqueezeNet

SqueezeNet is a deep neural network that facilitates the creation of a smaller network with fewer parameters while also maintaining an accuracy level that is similar to that of AlexNet. SqueezeNet has 50X fewer parameters than AlexNet while running 18 layers deep.

3.2.3. ResNet

The ResNet is a CCN based on deep architectures that have compelling accuracy and exhibits good convergence behavior. These models were developed by He *et al.* [6] and have managed to win first place in both the common objects in context (COCO) and the ILSRC classification challenge in 2015. The ResNet's architecture includes a number of stacked residual units paired with a different number of layers such as 1202, 152, 101, 50, 34, 18. However, the operations are bound to change, depending on the type of architecture [12]. All the residual units are made up of pooling, convolutional, and layers. While ResNet exhibits some similarities with VGG net [22], it runs eight times deeper compared to VGG [28]. The ResNet 18 is the ideal option based on its performance and depth. It is made up of a fully-connected layer with softmax, one average pooling, and five convolutional layers. The architecture of ResNet 50 packs 29 convolutional layers that are linked to the network with a fully-connected layer. In a bid to save the training time and computing resources, we chose ResNet 101, 50, 34, and 18 to help inform this study.

3.2.4. DenseNet

While increasing the depth of the convolutional neural network had its benefits, there existed the challenge of vanishing information about the gradient or the input when going through the layers [29]. The authors proposed a simple connectivity-based architecture to facilitate the maximum flow of information through the layers through both the forward computation and the backward computation. This solution provided a network that connected all the layers whereby one layer gets inputs from a preceding layer and passes it down to subsequent layers through its feature maps.

3.3. Evaluation

Evaluating the proposed method's performance can be done by using different metrics to compare the pre-trained models. On the other hand, the algorithms' quality can be evaluated by testing how well the learning algorithm will perform when given test [30]. The recall or sensitivity shows the accuracy level of the positive examples. It will also show the number of all the correctly labeled positive classes and can also be calculated using (1).

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

where,
TP (True Positives): the number of instances identified positively and correctly.
FN (False Negatives): the positive classes that are labeled as negative.

Accuracy is one of the popular metric used in evaluating the performance classification. During the evaluation stage, we calculated the accuracy after every 20 iterations. Accuracy is arrived at by calculating the percentage of the correctly classified samples. The Accuracy of a model can be calculated using (3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2)$$

Precision is true positives divided by a total of both positive and false positives. It can be calculated using (2). Precision deals with a model's correctness, for example, the algorithm's predictive power. It checks how precise a model is based on its predicted positives and the number of predicted positives that are actually positive.

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

The F-Score, on the other hand, is determined by the recall and the mean harmonic precision and can be calculated using (3). This metric is tasked with the analysis of positive class, meaning that a high value means that the model's performance is better on the positive class.

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (4)$$

4. RESULTS

This study assessed state-of-the-art pre-trained models (ResNet 120, ResNet 101, ResNet 50, ResNet 34, ResNet 18, squeezeNet and Vgg.net (16 & 19)) in helping classify tomato plant diseases using images. Additionally, the study compared the various CNN models based on the AUC, F-Score, specificity, sensitivity, precision, and accuracy for five iterations.

From the results, all the eight models exhibited similar performance across the used train-test set splits. The results on the pre-trained models in Table 1 shows that DenseNet-120 recorded the highest results with a precision of 99.81%, a recall of 100%, F1-score of 99.91%, and an accuracy of 99.69%. SqueezeNet, on the other hand, scored the lowest results with a precision of 99.85%, 98.62% recall, an F1-score of 97.67, with an accuracy of 96.54%. Based on individual metrics, Vgg-19 had the highest precision at 100%, while ResNet 34 had the lowest precision at 99.33%. As for the recall, DenseNet-120, resNet 101, and ResNet-50 had the highest recall at 100%, while ResNet-18 had the least recall of 99.22%. As for the F1-Score, ResNet-101, ResNet 50, and Vgg-19 had the highest F1-score of 99.92%, while ResNet 18 had the least F1-score of 99.22%. Finally, based on the accuracy metric, DenseNet 120 had the highest accuracy of 99.69%, while SqueezeNet had the least accuracy of 97.6%.

Table 1. Pre-trained model performance measurement

Performance Measures	Vgg-16	Vgg-19	Squeeze-net	Resnet-18	Resnet-34	Resnet-50	Resnet-101	Densenet-120
Training-Test: 80%-20%								
Precision	0.9984	1	0.9969	0.9968	0.9933	0.9983	0.9983	0.9968
Recall	0.9984	0.9984	0.9938	0.9875	0.9933	1	1	1
F1	0.9984	0.9992	0.9953	0.9922	0.9958	0.9992	0.9992	0.9984
Accuracy	0.9958	0.9922	0.9761	0.991	0.99	0.9864	0.9828	0.9923
Training-Test:60%-40%								
Precision	0.9976	1	0.9985	0.9944	0.9936	0.9961	0.9976	0.9985
Recall	1	0.9977	0.9862	0.9976	0.9968	0.9992	1	0.9985
F1	0.9988	0.9989	0.9767	0.996	0.9952	0.9976	0.9988	0.9985
Accuracy	0.9927	0.9911	0.9654	0.9836	0.9829	0.9787	0.967	0.9893
Training-Test: 50%-50%								
Precision	0.9968	0.9963	0.9987	0.9936	0.9961	0.9969	0.9981	0.9981
Recall	1	0.9994	0.9856	0.9891	0.9955	0.9988	0.9994	1
F1	0.9984	0.9978	0.9921	0.9913	0.9958	0.9978	0.9988	0.9991
Accuracy	0.9886	0.9857	0.976	0.9832	0.9881	0.9929	0.9949	0.9969

Based on the results from training the models from scratch as shown in Table 2, Vgg-16 posted the highest results achieving a precision of 99.83%, 99.83% recall, 99.81 F1-Score, and an accuracy of 99.84%. Squeezenet, on the other hand, posted the lowest results with a precision of 99.13%, a recall of 99.03, an F-

score of 99.27%, and an accuracy of 97.61%. Based on the individual metrics, Vgg-16, ResNet-18, ResNet-34, ResNet-50 and ResNet 101 had the highest precision at 99.83%. These were followed by denseNet-120 and Vgg-19 at 99.68%, while SqueezeNet came last at 99.51. In terms of Recall, DenseNet 120 had the highest at 99.84% followed by Vgg-16 at 99.83%, Vgg-19 99.68%, ResNet 18 99.67%, ResNet 34 99.67%, while SqueezeNet and ResNet 101 came last with 99.03% and 99.01% respectively. As for the F1-score, Vgg-16 had the highest score of 99.84%, while DenseNet-120 had the lowest F1-score at 99.29%.

The results of the performance of the algorithms based on both pre-trained models and models strained from scratch are similar to those of their respective models. The DenseNet-120 algorithm based on the pre-trained model (50-50) had the highest results in terms of accuracy, recall, precisiom amd f1-score. The Vgg-16 based algorithm that was trained from scratch (50%-50%), on the other hand, had the highest results based on accuracy, recall, precision and the F1-score metrics. The SqueezeNet algorithm had the lowest results based on both pre-trained models in terms of accuracy, recall, precision score, and F1-score.

Table 2. Performance measures models trained from scratch

Performance Measures	Vgg-16	Vgg-19	Squeeze-net	Resnet-18	Resnet-34	Resnet-50	Resnet-101	Densenet-120
Training-Test: 80%-20%								
Precision	0.9983	0.9968	0.9951	0.9983	0.9983	0.9983	0.9983	0.9968
Recall	0.9983	0.9968	0.9903	0.9967	0.9967	0.9934	0.9901	0.9984
F1	0.9983	0.9968	0.9927	0.9975	0.9975	0.9959	0.9942	0.9976
Accuracy	0.9958	0.9922	0.9761	0.991	0.99	0.9864	0.9828	0.9923
Training-Test:60%-40%								
Precision	0.9976	0.9976	0.9913	0.9984	0.9976	0.9968	0.9929	0.9968
Recall	0.9976	0.9976	0.9937	0.9952	0.9929	0.996	0.9968	0.9976
F1	0.9976	0.9976	0..9927	0.9968	0.9952	0.9964	0.9949	0.9972
Accuracy	0.9927	0.9911	0.9654	0.9836	0.9829	0.9787	0.967	0.9893
Training-Test: 50%-50%								
Precision	0.9974	0.9981	0.9925	0.9943	0.9962	0.993	0.9944	0.9943
Recall	0.9949	0.9969	0.9931	0.9962	0.9911	0.9917	0.99	0.9956
F1	0.9961	0.9975	0.9928	0.9981	0.9936	0.9924	0.9922	0.9949
Accuracy	0.9984	0.9875	0.9604	0.9804	0.9773	0.9688	0.9392	0.9859

5. DISCUSSION

DL has the potential and presents a great opportunity to further research regarding its application in using digital images to classify plant diseases. Having accurate and rapid models are needed to facilitate the early detection of plant diseases. For this study, we used a dataset housing nine classes of tomato diseases in addition to a healthy class. The dataset had a total of 18,160 images. The dataset was then divided into train-test set splits including 20-80 (20% of the dataset for training and 80% used for training), 40-60 (40% for training and 60% for testing), 50-50 (50% of the dataset for training and 50% for testing), 60-40 (60% for training and 40% for testing) and 80-20 (80% for training and 20% for testing). The study worked with eight state-of-the-art pre-trained and untrained CNN models together with their algorithms including Densenet 120, (Residual Network)ResNet 101, ResNet 50, ResNet 30, ResNet 18, SqueezeNet and Vgg.net (Vgg-19 and Vgg-16).

All the selected pre-trained models were then subjected to an evaluation based on the following metrics: F-Score, specificity, precision, and accuracy. The results show that DenseNet 120 had the best performance while SqueezeNet, posted the least performance despite being the deepest CNN model with 18 layers. This points towards a poor architecture even with its numerous layers. It is our hope that this proposed method will provide great breakthroughs and will be a critical contribution to the agricultural field.

The results obtained in this study are similar to those obtained in another study done by Barman *et al.* (2020) on pre-trained MobileNet and Self-Structured CNN architectures. These are used to identify the classification of diseases on citrus leaves from a smartphone image-based citrus disease dataset. The trained MobileNet and Self-structured CNN achieved an accuracy of 98% compared to our 99.81% accuracy achieved by our DenseNet CNN. However, the study makes a great contribution by creating the possibility of real-world application of the CNN by developing a smartphone-based plant disease classification.

One of the key limitations faced in the development of this study included the number of images that were used for the experiments. It would be interesting to use a set of images taken in a controlled environment and use those unknown images to test the model or if the models could classify images of diseases in a plant as soon as it starts presenting itself. A mobile application that will implement the DenseNet-120 CNN would be beneficial for farmers to use in diagnosing tomato and other plant diseases.

6.   CONCLUSION

In this study, we propose and test the use of DP in convolutional neural networks (CNNs) in the classification of tomato plant diseases. To achieve this, the study evaluated scratch and pre-trained versions of state-of-the-art CNNs using a number of performance metrics to compare their performance such as Densenet 120, (Residual Network)ResNet 101, ResNet 50, ReseNet 30, ResNet 18, SqueezeNet and Vgg.net (Vgg-19 and Vgg-16) in identifying plan disease from images of tomato leaves from the chosen dataset. Based on the results, all the models successfully classified all the nine diseases in the tomato leaves present in the class. The pre-trained DenseNet 120 CNN had the highest results with a 99.68% precision, 99.84% F-1 score, and 99.81% accuracy. This shows the effectiveness of using a combination of a CNN model with fine-tuning adjustment in classifying crop diseases. From the comparison, it was clear that using CNNs is quite advantageous in terms of profitable training performance, a faster convergence rate, and does not require preprocessing

7.   CONTRIBUTION AND FUTURE WORK

Food is critical to the survival of humanity. However, the production of high-quality food has been impeded by challenges such as disease. Fortunately, we can rely on new technologies to help with the detection and prevention of these diseases. For example, we can rely on CNN to increase our accuracy in determining tomato plant diseases. In this paper, we work on achieving higher accuracy in classifying different plant tomato diseases. It is our hope that our work will contribute significantly to the area of agricultural research. Achieving food sustainability and a healthy population will require such innovations. By studying how we can use these technologies with more accuracy, we want to make the world a better place for us and future generations.

REFERENCES

[1]   H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh, "Fast and accurate detection and classification of plant diseases," *International Journal of Computer Applications*, vol. 17, no. 1, pp. 31-38, 2011. https://doi.org/10.5120/2183-2754.

[2]   J.G. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems engineering*, vol. 172, pp. 84-91, 2018. https://doi.org/10.1016/j.biosystemseng.2018.05.013.

[3]   M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: detection and saliency map visualisation," In *Human and machine learning*, pp. 93-117, Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-90403-0_6.

[4]   C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson and H. Lipson, "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," *Phytopathology*, vol. 107, no. 11, pp. 1426-1432, 2017, doi: 10.1094/PHYTO-11-16-0417-R.

[5]   K.P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311-318, 2018. https://doi.org/10.1016/j.compag.2018.01.009.

[6]   K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90.

[7]   A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," In *Advances in neural information processing systems*, vol. 25, pp. 1097-1105, 2012. https://doi.org/10.1145/3065386.

[8]   C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9. https://doi.org/10.1109/cvpr.2015.7298594.

[9]   C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826. https://doi.org/10.1109/cvpr.2016.308.

[10] D. P. Hughes, and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, November 2015. Retrieved 4 March 2021, from https://arxiv.org/ftp/arxiv/papers/1511/1511.08060.pdf.

[11] Y. Kawasaki, H. Uga, S.Kagiwada, Hitoshi Iyatomi, "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks," In *International Symposium on Visual Computing*, 2015, pp. 638-645. https://doi.org/10.1007/978-3-319-27863-6_59.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, "Caffe: Convolutional architecture for fast feature embedding," In *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675-678. https://doi.org/10.1145/2647868.2654889.

[13] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic investigation on a robust and practical plant diagnostic system," In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 989-992. https://doi.org/10.1109/ICMLA.2016.0178

[14] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, 2016, doi: 10.1155/2016/3289801.

[15] S.P. Mohanty, D.P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, pp. 1419, 2016. https://doi.org/10.3389/fpls.2016.01419

[16] Nachtigall, Lucas G, Ricardo M. Araujo, and Gilmar R. Nachtigall, "Classification of apple tree disorders using convolutional neural networks," In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2016, pp. 472-476. https://doi.org/10.1109/ictai.2016.0078.

[17] Y. Lu, S. Yi, N. Zeng, Y. Liu, Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378-384, 2017. https://doi.org/10.1016/j.neucom.2017.06.023.

[18] G. Huang, Z. Liu, L. Maaten, K. Q. Weinberger, "Densely connected convolutional networks," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708. https://doi.org/10.1109/CVPR.2017.243.

[19] F.N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," arXiv preprint arXiv:1602.07360, 2016. https://arxiv.org/abs/1602.07360.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014. https://arxiv.org/abs/1409.1556v6.

[21] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Computational Intelligence and Neuroscience*, 2017, https://doi.org/10.1155/2017/2917536.

[22] J. Wang, L. Chen, J. Zhang, Y. Yuan, M. Li, and W. Zeng, "CNN Transfer Learning for Automatic Image-Based Classification of Crop," In *Image and Graphics Technologies and Applications: 13th Conference on Image and Graphics Technologies and Applications, IGTA 2018*, Beijing, China, April 8-10, 2018, vol. 875, pp. 319-392, 2018, https://doi.org/10.1007/978-981-13-1702-6_32.

[23] A.K. Rangarajan, Raja Purushothaman, Aniirudh Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia computer science*, vol. 133, pp. 1040-1047, 2018, doi: 10.1016/j.procs.2018.07.070. https://doi.org/10.1016/j.procs.2018.07.070.

[24] I. Khandelwal and S. Raman, "Analysis of transfer and residual learning for detecting plant diseases using images of leaves," In *Computational Intelligence: Theories, Applications and Future Directions-Volume II*, pp. 295-306. Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-1135-2_23.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, 115(3), pp.211-252, 2015. https://doi.org/10.1007/s11263-015-0816-y.

[26] U. Muhammad, W. Wang, S. P. Chattha, and S. Ali. "Pre-trained VGGNet architecture for remote-sensing image scene classification," In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1622-1627. IEEE, 2018. https://doi.org/10.1109/icpr.2018.8545591.

[27] A.F. Agarap, "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018. https://arxiv.org/abs/1803.08375.

[28] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers in neuroscience*, 2019, 13, p.95. https://doi.org/10.3389/fnins.2019.00095.

[29] F. Zhou, X. Li, and Z. Li, "High-frequency details enhancing DenseNet for super-resolution," *Neurocomputing*, 290, pp.34-42, 2018 https://doi.org/10.1016/j.neucom.2018.02.027.

[30] J. Zhang, C. Lu, X. Li, H.J. Kim, and J. Wang, "A full convolutional network based on DenseNet for remote sensing scene classification," *Math. Biosci. Eng*, 2019, 16(5), pp.3345-3367. https://doi.org/10.3934/mbe.2019167.

BIOGRAPHIES OF AUTHORS

Mohammad Amimul Ihsan Aquil is a Masters by Research candidate in Information Technology and a graduate research assistant in Universiti Teknologi Petronas. His research and professional interests include Artificial Intelligence, Data Science, Financial Engineering, Environmental Modelling and Neuropsychology. He graduated with first class from Universiti Utara Malaysia in Information Technology with Artificial Intelligence major in 2016. Before commencing his graduate study, Aquil worked at SBA Research, Austria. There, he focused on usable security applications for Software Engineering and Data Privacy Algorithms. He likes debate, programming and teaching.

Wan Hussain Wan Ishak is a Senior Lecturer at School of Computing, Universiti Utara Malaysia. He obtained his bachelor's degree in information technology and Master of Science in Information Technology from Universiti Utara Malaysia in the year 2000 and 2003 respectively. As an academician, his research interests include intelligent system, web-based content management system, and knowledge management. His works have been published in several international conferences, journals and won awards on national research and innovation competition.