International Journal of Application on Sciences, Technology and Engineering (IJASTE)
Volume 1, Issue 1, 2023. ISSN 2987-2499

Detection and Categorization of Tomato Leaf Diseases Using Deep Learning

Lih Poh LIN[1, a)] and Jia Shean LEONG [1, b)]

[1]*Department of Electrical and Electronics Engineering, Tunku Abdul Rahman University College, 53300 Kuala Lumpur, Malaysia*

[a)] Corresponding author: linlp@tarc.edu.my
[b)]1althea1990@gmail.com

Submitted: November-December 2022, Revised: January 2023, Accepted: February 21, 2023

Abstract. Tomato planting has been increasingly developed throughout the year. However, the yield from tomato farming is still heavily affected by diseases. Timely plant disease assessments can help farmers to control the spread of the disease and prevent major clusters of diseases. Conventional disease identification methods with manual inspections are inconsistent and inefficient, calling the need for artificial intelligence to aid tomato plant diseases identification. Convolutional neural network (CNN) is one of the most commonly implemented deep learning models in classification since it requires no data pre-processing, has a better convergence rate and generates decent training performances. CNN state-of-the-art architectures namely the VGG16, ResNet50 alongside a newer architecture MobileNetV2 were implemented in this study; and the performances of the various models in the classification of 10 categories of tomato leaf diseases were evaluated. The various CNN architectures were implemented under different training-to-testing ratios and with the absence/presence of data augmentation to investigate how their performances were affected. It was observable that the difference in loss and accuracy with or without data augmentation was not significant, likely due to the utilization of pre-trained models and the sufficiently large dataset. After training, the highest accuracy acquired was 90.19% for VGG16, 68.08% for ResNet 50 and 90.84% for MobileNetV2. At the training-to-testing ratio of 80:20, the shallow VGG-16 model presented the best performance with high accuracy and the lowest loss, as well as decent recall and precision, endorsing the feasibility to use a shallow model to achieve a promising classification of tomato leaf diseases.

INTRODUCTION

Background and Proposed Investigations

Tomato plantation is common in Malaysia, a study has shown that tomato cultivation has three years of breakeven period with an internal rate of return of approximately 26% [1]. Nonetheless, the yield of the tomato plant is greatly dependent on disease control. The existing and most widely used method to identify plant diseases is visual identification, where experts are trained to visually inspect plant diseases following many guidelines [2]. Timely plant diseases assessments can help farmers to control the spread of the disease and prevent major clustering. However, the inconsistency of manual inspection is driving the transition towards assessing plant diseases via artificial intelligence. In deep learning, a type of artificial intelligence, the data is passed through many layers (input, hidden, and output), and consists of voluminous nodes [3]. These nodes are attached to different weightage and coefficients to approximate the data to the result. Generally, the outputs of each layer will be the inputs of the next layer. The increasing number of layers gives rise to the idea of deep learning.

.

A type of deep learning, the convolutional neural network (CNN) has shown great promise in high speed and high accuracy image classification [4]. CNN classifies images by extracting the key features of the image instead of scanning the whole image pixel by pixel. The advantages of CNN include the exclusion of pre-processing of data, a

better rate of convergence and more favourable training performances [5]. There are three layers in CNN. Namely, convolution layer, pooling layer and fully connected (FC) layers. The convolution layer transforms the images into data via filters specified for different features. The pooling process acts as a noise filter by eliminating areas of an image with unimportant features. The data will be processed repetitively through many layers of convolution and pooling before going into the FC layer, which handles the final analysis of the images. Many CNN architectures have been invented over the years. Some of the most popular architectures are AlexNet, MobileNetV2, VGG16 net, and Residual Network (ResNet).

CNN-based solutions to identify diseases in vegetables and fruits have been reported in the literature [6]. A study was performed using CNN architectures AlexNet and VGG16 to classify tomato crop diseases [7]. It was observed that fine-tuning the mini-batch size, weight, and bias learning rate did not impact the classification accuracy. Furthermore, it was found that the image number significantly affects the performance, with 373 images producing the maximum accuracy. This finding is worth investigating to evaluate how a small dataset could generate a reliable classification performance. Other than that, a study was also conducted to compare different CNN models in classifying tomato plant disease [5]. The researchers fine-tune different CNN architectures including ResNet18, ResNet50, and AlexNet to classify tomato plant diseases. ResNet18 and ResNet50 achieved 99.06% and 99.15% accuracy respectively, followed by AlexNet which has an accuracy of 98.93%. Nonetheless, it was speculated that the good accuracy was mainly contributed by the large dataset. The dataset has over 54,000 images and has played a huge role in the training of the model, whom if absent, may not generate the respectable accuracy that was reported.

Based on the review of various deep learning models in the literature, the performance of the architecture normally goes up as the number of layer increase. Hence, a deeper network such as ResNet50 generally has better performance [8]. However, the training of a deeper network is more computational costly; while models like AlexNet and MobileNetV2 consume less computational power owing to their shallow architecture. Therefore, this work implemented a convolutional neural network (CNN)-based classification model to classify 10 types of tomato plant leaf diseases, with one of the objectives to evaluate the feasibility of using a shallower network to achieve comparable leaf diseases classification with less computing power. Model VGG16, ResNet50 and MobileNetV2 are selected as the architecture of interest. Besides, many of the previous works employed the training-to-testing dataset ratio of 80:20 by default and have always performed data augmentation. There is a lack of research that compared different CNN architectures under different dataset ratios alongside the presence/absence of data augmentation. To address the gap, this work investigated how, under different training conditions and with/without data augmentation, the performance of the CNN models are altered. CNN classifies inputs by extracting the key feature of each disease, skipping the need for manual-engineered feature extraction. Thus, it is anticipated that the research would yield a positive outcome in tomato leaf disease classification.

METHODOLOGY

Dataset

The dataset is obtained from the plant village. The datasets employed in the study consists of the following tomato plant diseases: leaf mould, early blight, late blight, target spot, mosaic virus, yellow leaf curl virus, bacteria spot, Septoria leaf spot and spider mites. The description of these diseases and a healthy tomato plant image is shown in Table 1. The total number of data in the dataset is 22930 images, categorized into 10 classes including healthy leaf. Table 2 shows the number of images per class. The balanced number of images per category is beneficial for the training of the deep learning model. The images are plucked tomato plant leaves with a consistent background and relatively consistent lighting, with the dimension of 256 pixels x 256 pixels and 96 dpi x 96 dpi. In traditional machine learning, image classification is usually done with features that are hand-crafted [29]. While in deep learning, the model learns from data without any hand-crafted features. However, to achieve reliable accuracy with the model, there needs to be a massive amount of data which makes CNN data-dependent. Hence, data augmentation is used as a solution. Online augmentation is employed in this research, in which the data are altered and adjusted in every training epoch. This artificially increases the number of different images the model is training on. The adjustment of the images includes shearing, zooming in and out of the images, and flipping the images horizontally. Online augmentation indicates that the CNN model will see an entirely different dataset for every epoch.

TABLE 1. CATEGORIES OF LEAF DISEASES

Categories	Description	Categories	Description
Leaf Mold 	A common disease usually caused by fungal [9]. At the early stage of infection, yellow spots appear at the front side, sometimes accompany by brown sporulation at the back. At the last stage, leaves roll up and dry.	Bacteria spot 	Symptoms include lesions forming on leaf parts. The lesion spreads to the rest of the leaves [15]. The spot could occur with or without yellowing. As the spots expand, the centres of previous spots could be sunken.
Early blight 	A common disease among potato and tomato plants that could be caused by pathogens such as Alternaria Alternata [10]. Small, brown lesions on tomato plant leave. The spots enlarge and form a bull's eye pattern. The surrounding area of the lesions turns yellow colour occasionally.	Septoria leaf spot 	Septoria leaf spot is also known as Septoria blight [15]. It is a common disease caused by a fungus named Septoria lycopersici. Although it is not fatal for tomato plants, it spreads and defoliates quickly. It weakens the tomato plants and the affected plant cannot grow fruit.
Late blight 	A deadly disease in cool and wet weather [11]. The leaf spots expand and white molds begin at the affected area margin. The leaves brown and shrivel within 14 days.	Spider mites 	Spider mites are in the family of Tetranychidae [16]. They attack plants like peppers, potatoes, and tomatoes. They live on the plant leaves and produce protective webs, which causes damage to plant cells.
Target spot 	A disease caused by the fungus, Corynespora Cassiicola [12]. It begins with tiny dark lesions that expand to form light brown lesions in concentric shapes.	Yellow leaf (Curl Virus) 	This disease is caused by the tomato yellow leaf curl virus [14]. It is one of the most destructive diseases of the tomato plant. The symptoms include leaves curling, the yellow edge of the leaf, tinier leaves than ordinary leaves.
Mosaic virus 	The affected area shows alternating colours of yellow and dark green [13]. the latter often appearing thicker and raised giving a blister-like appearance. The leaves often with pointed tips and some leaves might be twisted.	Healthy 	A healthy tomato plant leaf.

TABLE 2. NUMBER OF IMAGES PER CATEGORY

Categories	Total Images	Categories	Total Images
Bacterial spot	2127	Septoria Leaf Spot	2181
Early Blight	2400	Spider Mites	2176
Healthy	2407	Target Spot	2284
Late blight	2314	Mosaic Virus	2238
Leaf mould	2352	Yellow Curl Virus	2451

Transfer Learning and Performances Metrics

In order to achieve more efficient feature learning from the images of tomato leaf, transfer learning is employed in this study. All the CNN models for the tomato leaf disease classification system are acquired from the Keras Application library. Keras applications provide deep learning models with pre-trained weights, acquired from the training of these models on the ImageNet data. Rather than having the models to train from zero with random weight initialization, transfer learning allows the pre-trained models to adjust the features learned from a related dataset before being employed for training the new dataset, in our study, the 22930 images of tomato leaves [17]. This reduces the computational load in model training. The advantages of using the transfer learning technique are that it reduces the cost of acquiring and annotating the data. In turn, it solves the problem of model training with insufficient data.

This experiment uses four performance metrics, namely accuracy, loss, recall and precision to evaluate model performance. Accuracy is one of the most commonly used metrics in the machine learning model [18]. The main focus of accuracy is to measures the ratio of correct predictions (true positives and true negatives) over the total number of data that have been evaluated. Equation 1 shows the accuracy formula. Meanwhile, the loss is computed via a loss function, which is also known as a cost function [19]. It measures the difference between the actual output and the predicted output achieved via forward-propagation. Accuracy and loss are measured for the selection of the best CNN model. The selected CNN model is further analysed with recall and precision. Recall shows the proportion of actual positives that are recognized correctly, as shown in equation 2. Meanwhile, precision measures the positive identifications that are truly correct, as shown in equation 3. In the application of multiclass classification, one commonly used loss function is cross-entropy. The standard formula of it is shown in equation 4 [20].

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

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

Where TP is true positives, TN is true negatives, FP is false positives, FN is false negatives.

$$J_{cce} = -\frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M} y_m^k \times \log\left(h_\theta(x_m, k)\right) \quad (4)$$

Where
J_{cce} is the categorical cross-entropy loss,
K is the number of classes,
M is the number of examples in training,
y_m^k is the target label for the training example m for class k,
x_m is the input for training example m,
h_θ is the model with neural network weight θ.

CNN Architectures Description and Models Setup

Residual Network (ResNet): ResNet architecture offers solutions to the vanishing gradient problem and provides a good convergence rate and accuracy [5]. It is introduced in the ILSVRC 2015 challenge and won the contest with a 3.57% error rate [8]. ResNet relies heavily on its stacked residual units and these units are the building blocks of the network. Each unit comprises convolution, pooling, and layers. It has 3×3 filters, has more layers than VGG16, uses global average pooling instead of FC layers. The residual units were later updated and started using identity mappings and higher accuracy was obtained. Common ResNets include ResNet50 (50 layers) and ResNet101 (101 layers). Due to the number of layers in a typical ResNet, it poses computational load problems, making it computationally intensive to train a ResNet model.

VGG16: In the ILSVRC-2014 contest, VGG16 attained second place. It uses a small kernel size filter, decreasing the parameters within the convolutional layers and the model training time [8]. The network comprises 13 convolutional layers and 3 FC layers, with ReLu as its activation function. The model uses 3×3 stacked convolution layers and max-pooling to decrease the volume size in downsampling.

MobileNetV2: For neural networks to achieve high accuracy often demands high computational power. MobileNetV2 is developed by Google and its main contribution is its layer module, the inverted residual structure with a linear bottleneck. The performance of the module is tested on ImageNet classification. MobileNetV2 is designed for mobile uses and limited-resource environments. The model has been concluded to have highly memory-efficient and allow standard operations to be presented in all frameworks.

All the CNN models used in the experiments are acquired from the Keras Application library. Three architectures are trained to classify the 10 tomato plant leaf diseases. As mentioned, these architectures are MobileNetV2, VGG16 and ResNet50. The dataset is separated in different train-to-test ratios (i.e., 60:40, 70:30, 80:20, 90:10) and are all resized to 224 pixels x 224 pixels (the models image input size). The models have been trained with both augmented and non-augmented datasets 10 times (10 training epochs). Table 3 shows the properties used for all the architectures. The activation function used is softmax, the loss function is categorical cross-entropy and the optimiser is adam. The weights of the architectures are initiated as the weight from the ImageNet using the transfer learning technique. These weights are not trained and only the last few layers are trained.

In summary, Figure 1 shows the general flow of the tomato leaf disease categorization process. The deep learning model is capable to categorize the tomato leaves into 10 different classes.

TABLE 3. COMMON ARCHITECTURES PROPERTIES

Properties	Description
Batch Size	16
Weight Initialization	ImageNet
Activation function	Softmax
Loss Function	Categorical crossentropy

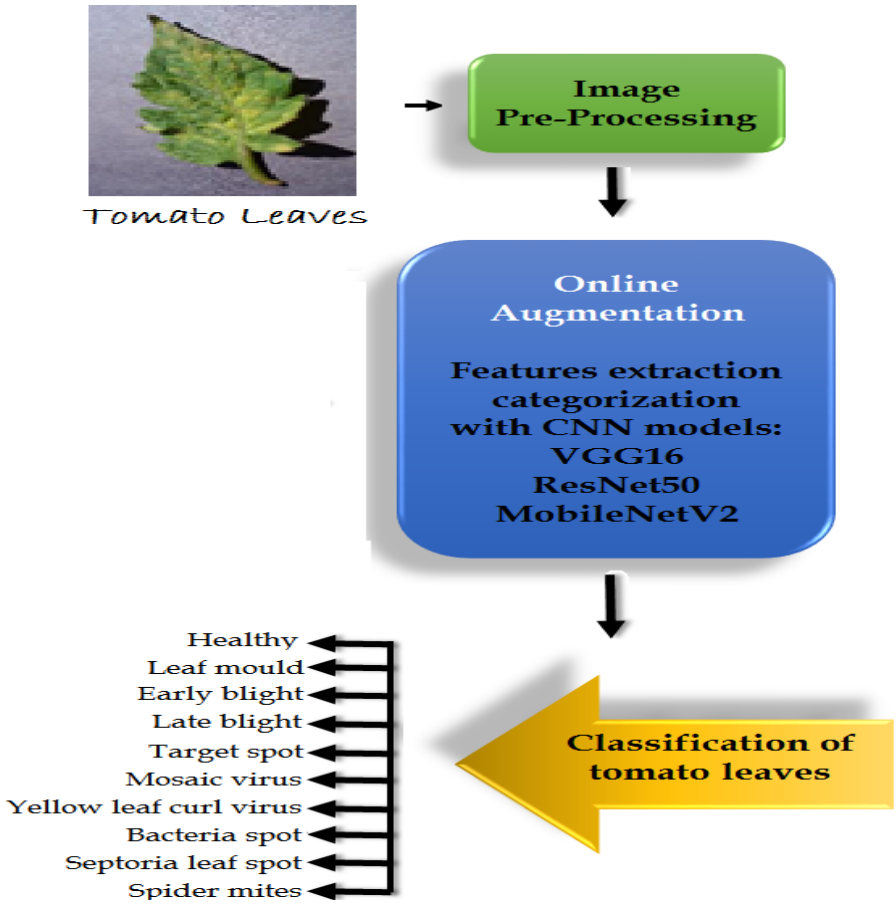


Figure 1. Block diagram of the tomato leaf disease classification system.

RESULTS AND DISCUSSION

Effect of CNN Architecture, Training: Testing Ratio and Data Augmentation on Classification Performances

CNN of three different architectures are tested and the resulting loss and accuracy of the models are recorded. Table 4 shows the testing result of all three models before training but with ImageNet weight initialized. As seen from the table, the accuracy is around 10% for most of the models, except an outlier of VGG16 which has an accuracy of 44.23% at an 80:20 ratio. The result manifests that the models with transfer learning alone could not produce good accuracy for the classification of tomato leaf diseases. Hence, there is a need to perform training to achieve better accuracy for the models.

TABLE 4. RESULTS OF MODELS WITHOUT TRAINING

	60-to-40		70-to-30		80-to-20		90-to-10	
	Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
VGG16	2.4963	7.21%	2.5597	10.40%	1.9467	44.23%	2.5275	8.96%
ResNet50	2.9846	9.78%	2.6263	9.34%	2.6957	11.15%	2.7541	10.30%
MobileNetV2	3.3257	9.77%	3.4542	8.43%	3.6757	12.28%	3.7804	11.07%

Subsequently, the model is trained with the dataset and generally, the objective of tomato leave classification is achieved as images are categorized into the different diseases with good accuracy. Table 5 shows the results of the

models trained without data augmentation at the 10th epoch. The loss is lowest with VGG16, followed by ResNet 50 and MobileNetV2 across different training-to-testing ratios. Furthermore, the accuracy of VGG16 and MobileNetV2 are consistently higher than ResNet50 over different train-to-test ratios. Table 6 shows the models trained with data augmentation. Comparatively, the loss of VGG16 is the lowest among all three models, followed by ResNet50, then MobileNetV2. It is observable that VGG16 generates a comparatively low loss with or without data augmentation The accuracy of MobileNetV2 is the highest among all, followed by VGG16 and ResNet50. ResNet50 has a significantly lower accuracy than the former two models.

TABLE 5. RESULTS OF MODELS WITH NO DATA AUGMENTATION AT 10TH EPOCH (AFTER TRAINING)

	60-to-40		70-to-30		80-to-20		90-to-10	
	Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
VGG16	0.5127	89.99%	0.7411	74.11%	0.5341	90.19%	0.7114	87.98%
ResNet50	3.6378	54.22%	1.5057	68.08%	1.6334	65.04%	1.6433	66.07%
MobileNetV2	7.4586	86.91%	6.2509	88.35%	4.9156	90.84%	5.0762	89.84%

TABLE 6. RESULTS OF MODELS WITH DATA AUGMENTATION AT 10^{TH} EPOCH (AFTER TRAINING)

	60-to-40		70-to-30		80-to-20		90-to-10	
	Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
VGG16	1.0521	81.52%	0.6583	87.00%	0.8116	89.91%	0.5607	87.83%
ResNet50	2.7264	56.28%	2.6858	54.08%	3.0351	47.76%	2.4948	56.04%
MobileNetV2	6.6677	85.43%	4.5897	89.47%	3.9392	90.82%	3.6979	91.41%

Based on Table 5 and 6, The 80:20 training-to-testing ratio which provides the highest accuracy is the best for model training. Nonetheless, the presence of data augmentation is observed to not generally affect the model performance. Table 7 shows the comparison between models that are trained with and without data augmentation. The "Changes with Aug" columns are calculated by deducting the non-augmented values from the augmented values. The data suggests that there is no significant overall improvement using data augmentation in model training. The reasons could be due to the use of pre-trained models and also the sufficiently large dataset (22930 images) employed in this research. However, this does not mean that data augmentation is worthless for model training since it allows the model to be trained on more data instead of repeating data, especially for small datasets.

One of the objectives of this work is to evaluate the feasibility of using a shallower network to achieve comparable leaf diseases classification performances with less computing power. The analysis of loss and accuracy suggests that VGG16 is the best model for the application of tomato plant diseases classification. Even though VGG16 has the second-best accuracy of 90.19%, it is not significantly inferior as compared to the MobileNetV2 which has an accuracy of 90.84%. MobileNetV2 is not the best option as it presents the highest loss alongside its high accuracy. The decent performance of VGG16 endorses the feasibility of employing a shallow network to achieve a respectable classification accuracy. The recall and precision of VGG16 are also analyzed to further complement its good performance in tomato leaf diseases detection. By fixing the training epoch to 10 and the train-to-test ratio to 80-to-20, the overall performance including accuracy, loss, recall and precision of VGG16 is presented in table 8. Overall, the performance of VGG16 is encouraging; it presents low loss and gives detection of high accuracy, recall and precision of about 90%

TABLE 7. PERFORMANCE COMPARISON BETWEEN TRAINING WITH AND WITHOUT DATA AUGMENTATION

	60-to-40				80-to-20			
	No Augmentation		**Changes with Aug**		**No Augmentation**		**Changes with Aug**	
	Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
VGG16	0.5127	89.99%	0.5394	-8.47%	0.5341	90.19%	0.2775	-0.28%
ResNet50	3.6378	54.22%	-0.9114	2.06%	1.6334	65.04%	1.4017	-17.28%
MobileNetV2	7.4586	86.91%	-0.7909	-1.48%	4.9156	90.84%	-0.9764	-0.02%
	70-to-30				**90-to-10**			
	No Augmentation		**Changes with Aug**		**No Augmentation**		**Changes with Aug**	
	Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
VGG16	0.7411	74.11%	-0.0828	12.89%	0.7114	87.98%	-0.1507	-0.15%
ResNet50	1.5057	68.08%	1.1801	-14.00%	1.6433	66.07%	0.8515	-10.03%
MobileNetV2	6.2509	88.35%	-1.6612	1.12%	5.0762	89.84%	-1.3783	1.57%

TABLE 8. OVERALL PERFORMANCE OF VGG 16 AT 10TH EPOCH (AFTER TRAINING)

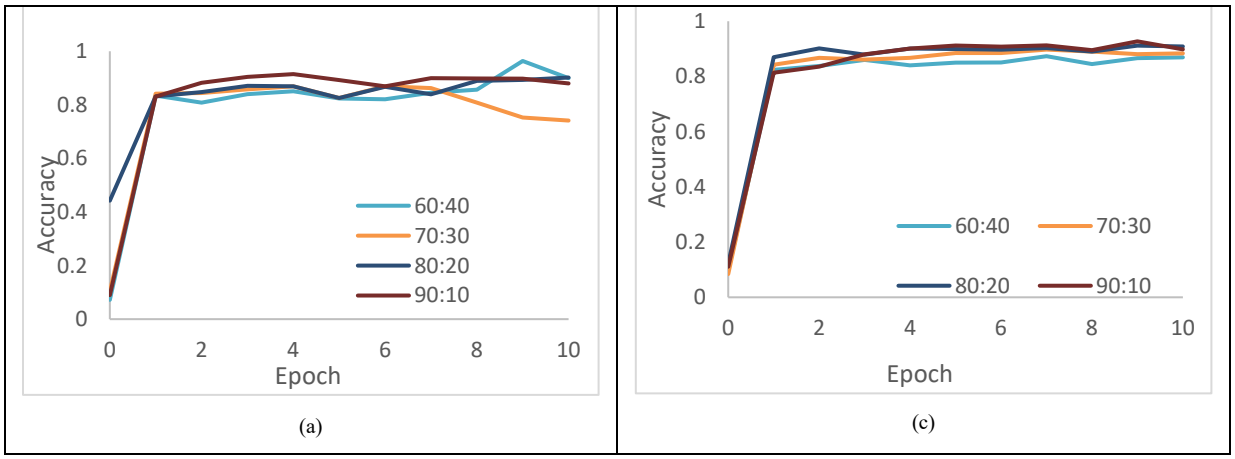
	VGG16			
	Loss	**Accuracy**	**Recall**	**Precision**
Without Data Augmentation	0.5341	90.19%	91.47%	90.35%
With DataAugmentation	0.8116	89.91%	89.99%	88.35%

CNN Models Convergence

Convergence indicates the progression into a steady state in which the CNN has sufficiently learned the features of training data within a certain margin of error. The absence of convergence suggests that the CNN model is not well-fitted. It is observed that the VGG16 model (Figure. 2(a)-(b)) and MobileNetV2 model (Figure 2(c)-(d)) converge and reach a stable state as the number of epochs increases, with or without data augmentation, attributed to the fact that the model is used to seeing the repeating data and have learned to identify features without data augmentation. The convergency is observed across all different training-to-testing ratios.

As well, the convergence of ResNet50 is evaluated. From Figure 3, it is noticed that the accuracy of ResNet50 fluctuates and does not converge within the 10 training epochs with or without data augmentation. Other than that, the accuracy of ResNet50 is constantly below 70% across all training-to-testing ratios. These two observations can be explained by the number of layers in ResNet50 that needs to be trained. The number of ResNet 50 layers is the highest of all three models, resulting in a more demanding training effort. A longer training time is required to improve the convergence of ResNet50.

In short, all the models have positive transfer learning and the accuracy reaches saturation rapidly at around the 4th epoch except for ResNet50. Model VGG16 is the preferable architecture given its decent accuracy and quick convergence across all training and testing conditions.



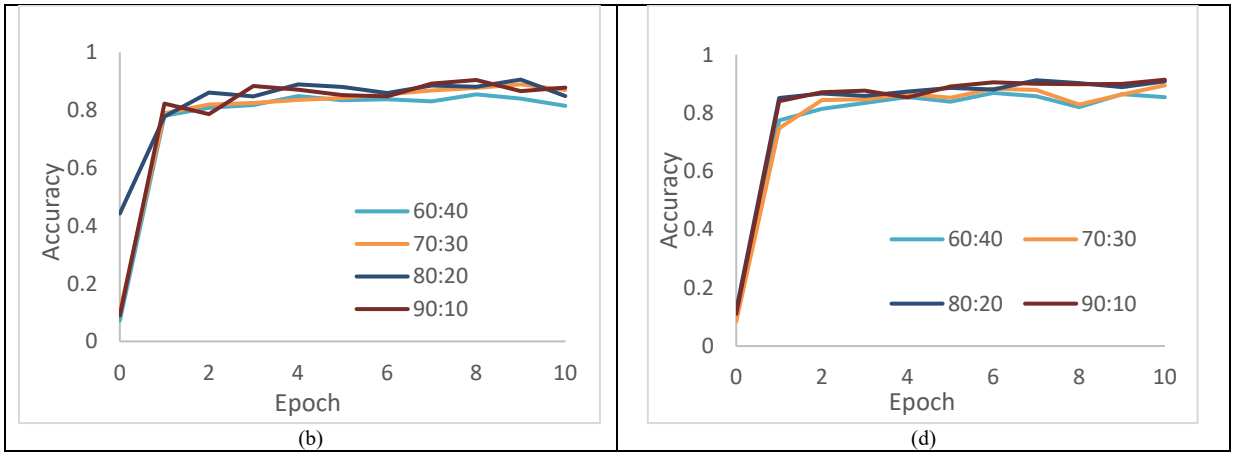


Figure 2. Convergence for VGG16 model (a) without data augmentation, (b) with data augmentation and MobileNetV2 model (c) without data augmentation (d) with data augmentation. Both models converge starting from the 4th epoch.

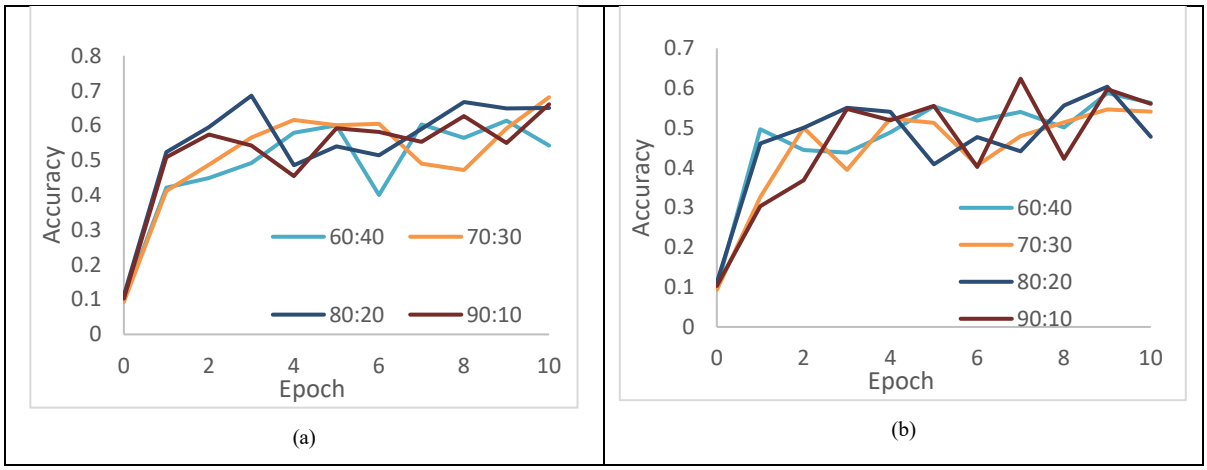


Figure 3. Convergence for ResNet50 model (a) without data augmentation, (b) with data augmentation. ResNet50 is a deep model that requires a longer training time to converge.

CONCLUSION AND FUTURE WORK

In this work, a CNN deep learning model based on VGG16, ResNet50 and MobileNet V2 architecture has been developed to classify tomato leaf diseases with high accuracy. CNN is a favourable deep learning method in classification since CNN does not require data pre-processing and has a better convergence rate and training performance. Three CNN models namely MobileNet, VGG16 and ResNet50 are evaluated to investigate how their performances are affected under different training conditions (training-to-testing ratio and the presence/absence of data augmentation) The highest accuracy acquired is 90.19% for VGG16, 68.08% for ResNet 50 and 90.84% for MobileNetV2. The proposed experiment manages to produce models with performance metrics that are comparable to previous work despite the limiting computational resources. The analysis of loss and accuracy suggests that VGG16 is the best model for the application of tomato plant diseases classification. Even though VGG16 has the second-best accuracy of 90.19%, it is not significantly inferior as compared to the MobileNetV2 which has an accuracy of 90.84%. MobileNetV2 is not the best option as it presents the highest loss alongside its high accuracy. The decent performance of VGG16 endorses the feasibility of employing a shallow network to achieve a respectable classification accuracy. Overall, the performance of VGG16 is encouraging; it presents low loss and gives detection of high accuracy, recall and precision rate of about 90%. It is also concluded that data augmentation may not always

improve the performance of a deep learning model, especially for applications that have a sufficiently large dataset, such as the one reported in this work. Future work should be driven to analyse the effect of altering the properties within the models trained (i.e. batch size, layers, weight and bias learning rate). Furthermore, more performance metrics such as sensitivity and F1 score should be employed for a more thorough evaluation of the models' performance.

ACKNOWLEDGMENTS

The authors acknowledge the facility provided by Tunku Abdul Rahman University College for the completion of this research.

REFERENCES

1. H. Rahim, M. Wahab, M. Amin, A. Harun and M. Haimid, Economic and Technology Management Rev., **12**, 41-53 (2017).
2. A. K. Mahlein, Plant Dis,. **100(2)**, 241-251 (2016)
3. Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., and Gao, X., Methods, **166**, 4-21 (2019).
4. W.Rawat and Z. H. Wang, Neural Comput., **29(9)**, 2352-2449 (2017).
5. V. Maeda-Gutiérrez *et al.*, Appl. Sci., **10(4)**, 1245 (2020)
6. M. Nagaraju and P. Chawla, Int. J. Syst. Assur. Eng. Manag., **11(3)**, 547–560 (2020).
7. A. K. Rangarajan, R. Purushothaman, and A. Ramesh, Procedia Comput. Sci., **133**, 1040–1047 (2018)
8. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, Comput. Electron. Agric.,**161**, 272–279 (2019).
9. J. Enya et al, J. Gen. Plant Pathol., **75**, 76-79 (2009).
10. P. S.Gorai, R.Ghosh, S.Konra, and N.C. Mandal, Biol. Control, **156**,104551 (2021).
11. M. Nowicki, M. R. Foolad, M. Nowakowska, and E. U. Kozik, Plant Dis., **96(1)**, 4–17 (2012).
12. K. Pernezny, P. Stoffella, J. Collins, A. Carroll, and A. Beaney, Plant Prot. Sci., **38(3)**, 81–88 (2018).
13. F. Eraslan, B. Akbas, A. Inal, and C. Tarakcioglu, Phytoparasitica, **35 (2)**, 150–158 (2007).
14. E. Glick, Y. Levy, and Y. Gafni, Plant Prot. Sci*.,* **45(3)**, 81–97 (2009).
15. K. M. Asit, K. M. Praveen, D. Subrata, and C.Arup, Agric. Res. Technol. Open Access J., **10(5)**. 00109-00117 (2017).
16. P. Tiftikci S. Kok, and I. Kasap, Biol. Control, **151**,104404 (2020).
17. M. Hussain, J. J. Bird, and D. R. Faria, Adv. Intell. Syst. Comput., **840**, 191–202 (2019).
18. M. Hossin and M. Sulaiman, Int. J. Data Min. Knowl. Manag. Process, **5(2)**, 01–11 (2015).
19. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, Smart Innov. Syst. Technol., **195**, 21–30 (2021).
20. Y. Ho and S. Wookey, IEEE Access, **8**, 4806–4813 (2020).