International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.141, September 2026

Performance and Computational Complexity Analysis of Pre-Trained Deep Learning Models for Citrus Fruit Disease Classification

Nur E. Jannatul Farjana
Dept. of Computer Science and Engineering
International University of
Business Agriculture and Technology
Uttara, Dhaka-1230, Bangladesh

Abdullah Miraz
Dept. of Computer Science and Engineering
International University of
Business Agriculture and Technology
Uttara, Dhaka-1230, Bangladesh

Krishna Das
Dept. of Computer Science and Engineering
International University of
Business Agriculture and Technology
Uttara, Dhaka-1230, Bangladesh

ABSTRACT

Citrus fruit diseases are a real threat affecting agricultural productivity and can lead to losses that are hard to recover if they are not detected and managed promptly. This research focuses on using transfer learning and deep learning architectures to classify citrus diseases to reduce agricultural losses and increase crop productivity. The study uses a dataset of 1,463 high-resolution images of citrus fruits and leaves that fall under these classes: canker, black-spot, greening, and healthy. The images were processed for analysis using a pre-trained Neural Network (CNN) such as MobileNetV2, InceptionV3, ResNet50, and VGG19. Transfer learning was used to fine-tune these models on the task of disease classification, which depends on features previously learned from larger datasets. Model performance was evaluated through accuracy, precision, recall, F1 score, and computational efficiency metrics. Out of the four models analyzed in this research study, MobileNetV2 attained the highest accuracy at 97.29% while ResNet50 did not perform well with an accuracy value of 58.21%. This emphasizes the importance of the selection of models. This research showcases how transfer learning and deep learning can bring scalability into automated systems for managing citrus diseases providing better accuracy, and quicker diagnosis times.

General Terms

Deep Learning, Image Classification, Pre-Trained Models

Keywords

Transfer Learning, Convolutional Neural Networks, Citrus Disease Classification, Performance Metrics, Computational Complexity, MobileNetV2

1. INTRODUCTION

Agriculture is one of the most important factors in Bangladesh as the national economy directly depends on it. Agriculture aims to increase food production and quality while reducing expenses and increasing profit [1, 2, 24]. Due to population growth and climate change, Bangladesh's agricultural sectors are using novel technology to increase food production. Precision agriculture (PA) is an advanced technology that provides modern techniques for optimizing farm production [30, 16]. Precision agriculture can be used in many ways to help farmers, such as spotting plant pests, detecting weeds, forecasting crop yields, and identifying plant diseases, among others [4].

Bangladesh's citrus fruit production increased at an average yearly rate of 5.61% from 19,163 tonnes in 1973 to 175,325 tonnes in 2022 [23]. This research was carried out in many citrus farms in Sylhet, Bangladesh from November to December 2014 to investigate the density and severity of diseases among various citrus species. Also an extensive analysis was performed on 560 plants belonging to seventeen different citrus species, but no symptom could be detected. The survey revealed that canker, scab, dieback, and greening were the major diseases in the area [8, 24]. Early symptoms of plant diseases are important for minimizing risks and reducing losses for farmers. In recent years deep learning has found widespread usage in countless domains including agriculture. For identifying diseases of plants, it is well known and recognized that CNNs are the most efficient deep learning technique [8]. Some of these architectures including AlexNet, GoogLeNet, VGG16, VGG19, InceptionV3, DenseNet, etc. are used for detecting and identifying plant diseases using CNN.

1.1 Context and Background

Being a developing country, crop diseases in Bangladesh negatively impact the economy and food resources [11, 19]. Specially, Bangladesh is a nation that is greatly dependent on agriculture such as paddy, corn, wheat, etc. The weather of Bangladesh is suitable for crop production but in recent years, crop diseases have reduced food production greatly. Reducing crop diseases is one of the important goals for the Government [11]. To capture the images, it can be brought together modern electronic devices with high-quality cameras and then use models such as Machine Learning or Deep Learning to process the image [24, 12]. Early detection of the disease through the identification of diseased areas in plants using advanced digital image processing methodologies could help in treating it effectively as well. In Bangladesh, 39,678 acres were used for crop cultivation and other agricultural purposes based on statistics done in 2019–2020 [15, 11].

1.2 Research Motivation

Deep learning models and supervised machine learning are used in the study that has already been done on the classification of citrus diseases from images [2, 8, 19]. AlexNet has 8 layers, while VGG-19 has 47 layers, both optimized using Stochastic Gradient Descent with Momentum (SGDM) [8]. However, it has high computational costs, significant memory requirements, longer training times, and suboptimal performance on small datasets. Also, they did not do a complexity analysis of those architectures. In this paper, a system is proposed that can effectively classify citrus fruit diseases and perform the complexity of the CNN architectures.

1.3 Objectives and Scope

The objectives and scope of this work include:

- To determine the effectiveness of different techniques for disease identification, CNN architectures are employed to classify citrus fruit diseases (InceptionV3, MobileNetV2, VGG19, ResNet50).
- Compare the models by doing performance analysis and complexity analysis.

1.4 Significance of Research

This research is done using transfer learning approaches to implement pre-trained models to classify citrus fruit diseases. Evaluating the performance metrics and complexity metrics, it will determine the most efficient architectures of the Convolutional Neural Network (CNNs).
The structure of this paper is as follows. Section 2 presents the literature review. In Section 3, the proposed methodology is discussed in detail. In Section 4, all experiments are conducted and the models are compared using some parameters, also addressing the computational complexity of the models. Section 5 presents the conclusion.

2. LITERATURE REVIEW

The people of Bangladesh heavily rely on different types of agriculture. According to statistics, about 87% of people directly or indirectly depend on agriculture [27]. Researchers have been studying the methods to detect plant diseases early using Deep Learning. In the following part, some papers have been reviewed related to this work and presented the findings of those papers:
Ahmed and Muqit [2] designed a Deep Neural Network model for automatically detecting citrus fruit and leaf diseases. The research is centered on employing deep learning techniques in the agricultural sector. This means that it intends to build a framework that can identify illnesses that affect oranges, both fruits and leaves. The proposed model consists of three main components: data acquisition, data pre-processing, and the application of CNN. They used 2293 images from PlantVillage and got an accuracy of 94.55%. However, they used only 213 images for each class which is not sufficient.
Balafas et al. [4] presented a detailed comparative analysis of ML and DL to detect plant leaf diseases. In this paper, a comparative study is carried out regarding the intrinsic performance of deep learning (DL) about machine learning (ML), especially towards plant leaf diseases with a focus on citrus trees. To categorize the disorders, they submitted their collected data into three well-known machine learning (ML) models: Support Vector Machine (SVM), Random Forest (RF), and Stochastic Gradient Descent (SGD); moreover, they fed their data into three decision-making (DL) models: InceptionV3, VGG16, and VGG19.
For identifying the diseases in citrus, Elaraby et al. [8] used a deep learning approach. The problem of optimizing a deep learning strategy to quickly and precisely identify illnesses in citrus crops is discussed in their study. The goal of the project is to use the deep learning models VGG16, VGG19, and InceptionV3 to better classify illnesses, including Black Spot, Canker, Greening, and Melanose.
A mobile-based system for detecting plant leaf diseases using deep learning is written by Sujatha et al. [30] that mainly addresses the plant diseases present in the agriculture sector and also highlights the diseases that cause significant crop production losses every year, providing a smartphone application that takes advantage of deep learning algorithms to diagnose diseases on plant leaves.
Detection of Bangladeshi-produced plant disease using a transfer learning based on deep neural model is studied by Hasan et al. [11, 3]. This paper addresses the early detection of plant diseases using a transfer learning approach with deep neural networks, focusing on Bangladeshi crops [3]. The research classified the plant diseases which are very common in Bangladesh by using pre-trained models such as AlexNet and VGG19. They trained the models using transfer learning on a plant leaf-specific dataset containing 18 categories [3].

3. PROPOSED METHODOLOGY

The dataset was thoroughly collected and preprocessed for this work to make it appropriate for image processing. Keras and TensorFlow were used for the experiment to utilize their reliable features, enabling developers to develop and deploy deep learning models. Various advanced Convolutional Neural Network architectures were tested, which include VGG19, ResNet50, InceptionV3, and MobileNetV2 due to their high accuracy in image classification and detection. The detailed workflow of this research is given in Figure 1.

3.1 Dataset Overview

A dataset of 1,463 images has been collected from PlantVillage and Kaggle that holds the four classes: canker, blackspot, greening, and healthy. These are the main diseases of the citrus plant which are kept in different folders. This program applies labels to the images indicating where disease is presented and what type of disease, as well as healthy samples for comparison.

3.1.1 Dataset Preparation. Before implementing the models, the dataset needs to be prepared for using in the design and evaluation

Table 1. Summary of Related Literature on Citrus and Plant Disease Detection

No.	Author	Year	Class Types	Dataset Size	Classifier Architecture	Accuracy
1	R. Sujatha et al. [30]	2021	Black spot, Canker, Greening, Melanose, Healthy	609 images	LSVM, SGD, RF, InceptionV3, VGG16, VGG19	89.5%
2	Khattak et al. [15]	2021	Black spot, Canker, Scab, Greening, Melanose, Healthy	2293 (Fruits+Leaves)	Custom CNN + Softmax	94.55%
3	A. Elaraby et al. [8]	2022	Anthracnose, Black spot, Canker, Scab, Greening, Melanose	759 (Fruits+Leaves)	AlexNet, VGG19	94.0%
4	S. Parez et al. [21]	2023	Multiple plant pathological conditions	Plant leaves	Lightweight CNN + Softmax	99.0%
5	Dehuan Luo et al. [18]	2023	Anthracnose, Canker, Melanosis, Scab, Brown Spot, Leaf Miner	3202 images	Light-SA YOLOv8	92.6%
6	Madhusudan G. et al. [17]	2023	Black spot, Melanose, Canker, Greening, Healthy	609 RGB images	CNN, ResNet152V2, DenseNet121, InceptionResNetV2	98.0%
7	Poonam Dhiman et al. [6]	2023	HLB, Black spot, Melanose, Canker, Scab	2950 images	CNN + LSTM	98.25%
8	W. Gómez-Flores et al. [10]	2024	HLB, Greasy spot, Deficiencies, Citrus Mite, Red scale	953 images	DNN, Transfer Learning	Not reported
9	Nouman Butt et al. [5]	2024	Canker, Citrus scab, Black spot	2184 images	SVM, KNN	99.6%
10	J. Shermila et al. [28]	2024	Fruit blight, Greening, Scab, Melanoses	759 images	Custom CNN + LSTM	96.0%
11	A.K. Saini et al. [25]	2024	Anthracnose, Melanose, Brown Spot	759 images	SADCNN-ORBM	99.76%

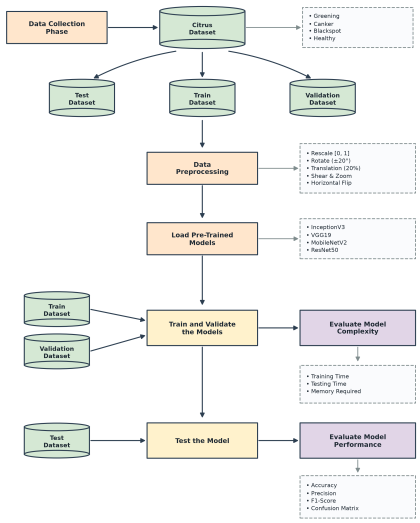

Fig. 1. Proposed Methodology

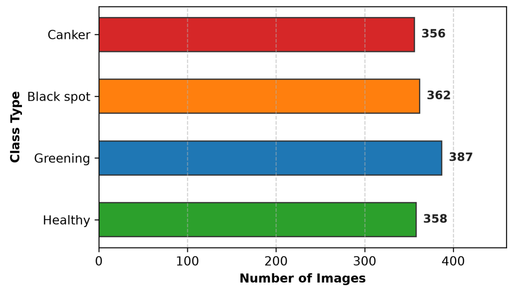

Fig. 2. Number of Images in Each Class

Fig. 3. Sample Images of Dataset

of deep learning models for disease classification. The steps of dataset preparation are given below:

—**Data Loading**: The dataset is broken down into directories, each of which contains images of a given class of citrus fruit disease. The paths to each of these images are stored along with their labels (type of disease) in a DataFrame, which is convenient for manipulation and analysis. Figure 2 shows the information about the citrus disease dataset that is used for this research work.

—**Initial Visualization**: A bar graph has been made representing the image distribution in different disease classes. This visualization helps to see how the dataset is behaving with its balance and has any type of class imbalance issue in the current system, or not. Figure 3 illustrates the sample images of citrus fruit disease dataset.

—**Data Splitting**: Dataset needs to be split into training, validation, and testing subsets to facilitate the model training. Datasets split up with 80% for training, 10% for validation, and 10% for testing. This split allows the model to have data with which it can learn and validation set where it can be tuned hyper-parameters without ending up over-fitting. Figure 4 shows a pie chart that illustrates the distribution of the dataset for training, validation, and testing.

3.1.2 Dataset Pre-processing. A special function has been designed that handles the pre-processing of training, validation, and testing datasets. The first step of pre-processing is image augmentation. Image augmentation has been done following these methods:

—**Rescaling**: The pixel values of the images are scaled into [0, 1].

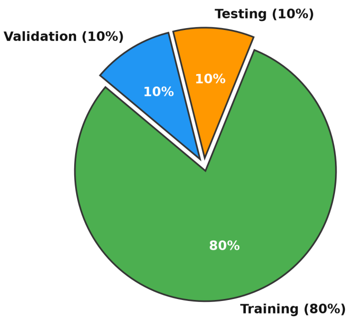

Fig. 4. Dataset Distribution for Training, Validation and Testing

—**Rotation**: Rotate the countered image any random degree between $[-20^{\circ}, 20^{\circ}]$.

—**Translation**: The image changes are performed at a random maximum of 20% pixels horizontally and vertically.

—**Shearing**: Random sheared the images.

—**Zoom Range**: Random zoom ranges to the images.

—**Horizontal Flipping**: Images are flipped randomly in the horizontal (left, right) direction. Images are loaded using a batch size of 32, and all the images from the dataset are resized into 224×224 pixels to ensure a fixed size for models. The test data is not shuffled to maintain the order of the images, which is important for evaluation purposes.

3.2 Transfer Learning

In deep learning, transfer learning is a method where the model trained for one task is used as a starting point in order to solve another related or similar problem [7]. This speeds learning and boosts performance on a second, possibly under-sampled issue by capitalizing the information obtained from a job. Transfer learning can be used for CNN's pre-trained models, feature extraction, and fine-tuning.

3.2.1 Pre-Trained Models. Transfer learning allows using CNN's pre-trained models which show better accuracy, even on medium-sized datasets [3]. As these models are already trained on benchmark datasets, they cost less and require less computational power. It also reduces training time and provides high accuracy.
Figure 5 presents the workflow flowchart of pre-trained models for better understanding.
Four pre-trained models have been used in this research: VGG19, MobileNetV2, InceptionV3, ResNet50. The working process of the pre-trained model is almost the same. After loading the base models, custom layers need to be added for this task. Then the models are created by giving input and output layers.

—**VGG19**: VGG19 model is famous for its simplicity and consistent architecture. It contains 19 trainable layers, deeper than many previous models. The network consists mainly of 3×3 convolutions, which is a quite new field since larger filters were still better than those at the time. First, the VGG19 model is pre-trained on the ImageNet dataset [26] but without the top classification, which is utilized as the base model to enable robust feature extraction [13, 29]. The layers of the base model are frozen, which prevents them from being trained further and enables them to use their learned features. Then a Dropout layer with a dropout rate of 40% is applied to prevent overfitting by setting a fraction of the input units to zero at random during training [9, 20, 31]. The output is then transformed into a one-dimensional vector suitable for the fully connected layers using a flattened layer [32]. A dense layer with 512 units and ReLU activation is added as the second fully connected layer. Another Dropout layer, with a dropout rate of 20%, follows the second dense layer. Finally, the output layer is a dense layer with 4 units and a softmax activation function, which gives the class probabilities for the classification task [22]. Table 2 presents the architecture of VGG19 in detail.

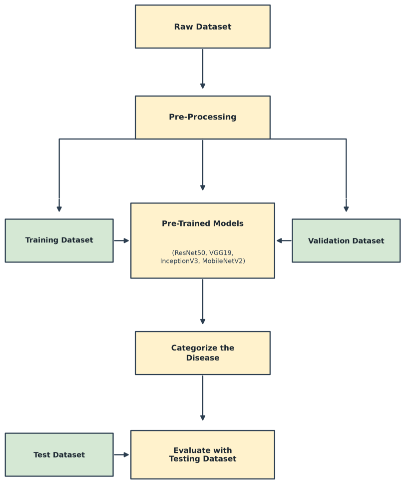

Fig. 5. Flowchart of Pre-Trained Models

Table 2. Layer Architecture and Parameter Details of Fine-Tuned VGG19

Layer (Type)	Output Shape	Parameters
VGG19 (Functional Base)	(None, 7, 7, 512)	20,024,384
dropout_1 (Dropout 40%)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense_1 (Dense, ReLU)	(None, 512)	12,845,056
dropout_2 (Dropout 20%)	(None, 512)	0
dense_2 (Dense, Softmax)	(None, 4)	2,052
Total Parameters:		32,871,492
Trainable Parameters:		12,847,108
Non-Trainable Parameters:		20,024,384

—**MobileNetV2**: MobileNetV2 Model (with pre-trained weights on ImageNet) is implemented and suited for plant classification using TensorFlow's Keras API [14]. By freezing each of the layers belonging to the base model, it conserves learned features from this original network while replacing only the final layers for task-specific customization. Dropout regularization and dense layers help in feature extraction and classification with an input image size of 224×224 pixels in RGB. The model is built with the Adam optimizer trained over some epochs and tested for its accuracy in a test set. Table 3 presents the architecture of MobileNetV2 in detail.

—**InceptionV3**: InceptionV3 is a CNN architecture for high performance image classification. The study uses InceptionV3, pre-trained on ImageNet, as the base model, with its top classification layers removed to facilitate plant-specific disease classification. The flattened output of the model is then fed to two dense layers with ReLU activation and dropout for regularization

Table 3. Layer Architecture and Parameter Details of Fine-Tuned MobileNetV2

Layer (Type)	Output Shape	Parameters
MobileNetV2 (Base)	(None, 7, 7, 1280)	2,257,984
dropout_1 (Dropout 40%)	(None, 7, 7, 1280)	0
flatten (Flatten)	(None, 62720)	0
dense_1 (Dense, ReLU)	(None, 512)	32,085,760
dropout_2 (Dropout 20%)	(None, 512)	0
dense_2 (Dense, Softmax)	(None, 4)	2,052
Total Parameters:		34,345,796
Trainable Parameters:		32,087,812
Non-Trainable Parameters:		2,257,984

before classifying it using a softmax layer. Table 4 presents the architecture of InceptionV3 in detail.

Table 4. Layer Architecture and Parameter Details of Fine-Tuned InceptionV3

Layer (Type)	Output Shape	Parameters
InceptionV3 (Base)	(None, 5, 5, 2048)	21,802,784
flatten (Flatten)	(None, 51200)	0
dense_1 (Dense, ReLU)	(None, 512)	26,214,912
dropout_2 (Dropout 20%)	(None, 512)	0
dense_2 (Dense, Softmax)	(None, 4)	2,052
Total Parameters:		48,019,748
Trainable Parameters:		26,216,964
Non-Trainable Parameters:		21,802,784

—**ResNet50**: The ResNet50 model is imported from TensorFlow Keras applications. This model has been configured to not include the top classification layers and it is difficult to train a very deep neural network on limited amounts of data. Keeping the base model as non-trainable prevents over-fitting when fine-tuning in small data. These custom top layers are a flattened layer and a dense layer after it. A dropout of 0.5 and an output that outputs in the softmax function for 4 classes. Adam optimizer is used with a learning rate of 0.0001, and the categorical cross-entropy of losses is considered to evaluate computing performance for this model. The model is then scored on a test set. Table 5 presents the architecture of ResNet50 in detail.

Table 5. Layer Architecture and Parameter Details of Fine-Tuned ResNet50

Layer (Type)	Output Shape	Parameters
ResNet50 (Base)	(None, 5, 5, 2048)	23,587,712
flatten (Flatten)	(None, 100352)	0
dense_1 (Dense, ReLU)	(None, 512)	51,380,736
dropout_2 (Dropout 50%)	(None, 512)	0
dense_2 (Dense, Softmax)	(None, 4)	2,052
Total Parameters:		74,970,500
Trainable Parameters:		51,382,788
Non-Trainable Parameters:		23,587,712

3.2.2 TensorFlow. TensorFlow is a well-known open-source machine learning framework developed by the Google Brain Team. It offers a holistic environment to create and implement machine learning models. TF is built for production on environments like CPUs, GPUs, and TPUs, making it suitable for developers or researchers. Due to comprehensive support for both production and research by TensorFlow, it is the preferred platform selected among other models for a multitude of machine learning as well as artificial intelligence applications. The architecture of TensorFlow is given in Figure 6.

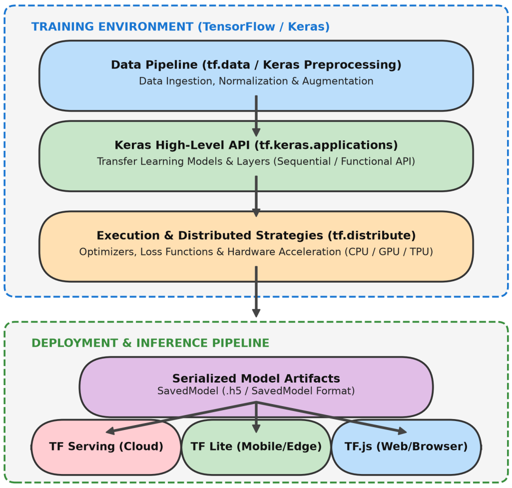

Fig. 6. Architecture of TensorFlow

Figure 6 provides a diagram of the components and connections for both training and deployment of TensorFlow.

—**Keras**: Keras is a high-level library developed in Python for building ML models using TensorFlow. It serves as a high level API for working with deep learning models to be used by machine learning frameworks such as TensorFlow. Keras allows users to experiment with all sorts of different neural network architectures quickly and prototype rapidly.

3.3 Optimization Process

In machine learning and deep learning, the process of optimization refers to adjusting a model's parameters such that it satisfies a few conditions, which are represented via loss function. The optimization of this research has followed the steps below:

—**Adam Optimizer**: The models are trained using Adam optimizer, which is an adaptive learning rate optimization algorithm that uses methods from both AdaGrad and RMSProp.

—**Learning Rate**: This is the rate which will determine how big of a step is taken for each iteration when moving toward the minimum point in the loss function ($\eta = 0.0001$).

—**Loss Function**: Categorical cross-entropy loss function is used here. This loss function is ideally used for multi-class classification problems where the target variable can be inferred from a categorical/dependent column:

$$\mathcal{L}_{CCE} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{1}$$

where N is the mini-batch size, $C = 4$ is the number of target classes, $y_{i,c} \in \{0, 1\}$ is the true label, and $\hat{y}_{i,c}$ is the model predicted probability for class c.

—**Training Process**: The model is trained using the fit method, which takes the training data generator, the number of epochs (50), and the validation data generator. The training process updates the weights of the trainable layers to minimize the loss function.

—**Layer Freezing**: The base layer of each model layer is frozen to keep the pre-trained weights as it is and train only the newly added dense and dropout layers.

3.4 Model Evaluation

Model evaluation is a process that provides the performance of a trained model onto unseen data. The purpose is to understand how well the model generalizes into new, unknown data. Models have been evaluated using two methods: performance analysis and complexity analysis.

3.4.1 Performance Metrics. For the proposed model, some parameters are used to evaluate the performances and compare their results. The metrics are used as follows:

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

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

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$F1\text{-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

3.4.2 Complexity Metrics. The complexity analysis of a model involves evaluating its computational and memory requirements, which is essential for understanding its scalability and feasibility for deployment. The metrics of complexity analysis are training time, required memory, and inference time.

3.5 Research Tools

The research work is fully conducted on Google Colab, so there is no need for external devices for this work. Additionally, the cloud-based nature of Colab allows for scalable computations and flexibility in handling different project requirements, ensuring there are no obstacles or limitations during the research process. Transfer learning approach has been implemented using Python Programming Language.

4. RESULT AND DISCUSSION

This part includes a detailed discussion of the results and findings throughout the research. The most effective model will be evaluated using two analyses: Performance analysis and Complexity analysis.

4.1 Performance Analysis

The performance analysis was done using metrics such as precision, accuracy, recall, and confusion matrix. The precision of VGG19, ResNet50, InceptionV3, MobileNetV2 was 84.82%, 66.66%, 97.21%, 98.56%. In this case, MobileNetV2 showed the highest precision and ResNet50 showed the lowest precision. In this way, MobileNetV2 achieved an F1 score of 98.54% which is the highest F1 score among all the models. This remains the same for the recall percentage. On the contrary, ResNet50 achieved the lowest value in all performance metrics in this experiment. InceptionV3 showed satisfactory value for all metrics. Table 6 presents performance metrics for four pre-trained models: MobileNetV2, InceptionV3, ResNet50, and VGG19, evaluating Precision, Recall, and F1 Score as percentages.

Table 6. Performance Comparison Across Fine-Tuned Models on the Unseen Test Set

Model Name	**MobileNetV2**	**InceptionV3**	**ResNet50**	**VGG19**
Precision (%)	98.56	97.21	66.66	84.82
Recall (%)	98.54	97.30	62.02	81.31
F1-Score (%)	98.54	97.19	58.21	81.17

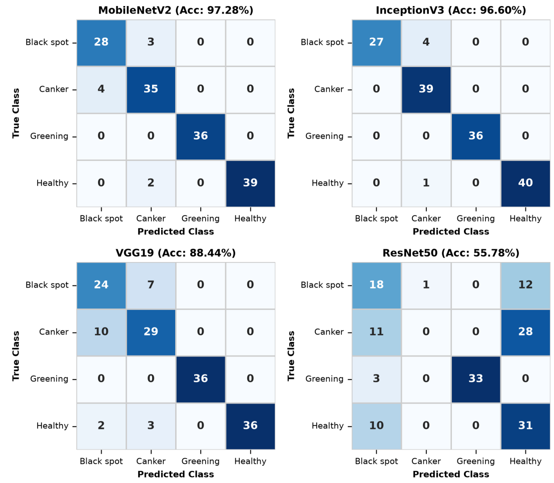

Fig. 7. Multi-class confusion matrices across all four fine-tuned CNN architectures on the test partition.

4.1.1 Accuracy Analysis. Each model's accuracy and loss during testing, training, and validation, have been evaluated to compare their performances. High accuracy indicates successful data capture, while low training loss indicates good performance and shallow loss may indicate overfitting, where the model learns noise rather than patterns. The training accuracy of MobileNetV2 and InceptionV3 is superior to other models. This remains the same for validation and test accuracy. MobileNetV2 produced the lowest loss value. On the other hand, ResNet50 has the highest loss value among all models. Table 7 presents evaluation metrics for models, including MobileNetV2, InceptionV3, ResNet50, and VGG19, training, validation, and testing phases, including accuracy and loss.

Table 7. Accuracy and Loss Metrics Across Training, Validation, and Test Phases

Model	**Training**		**Validation**		**Testing**	
	Acc. (%)	**Loss (%)**	**Acc. (%)**	**Loss (%)**	**Acc. (%)**	**Loss (%)**
MobileNetV2	95.80	10.97	94.55	12.80	97.29	5.29
InceptionV3	95.89	10.60	97.95	6.72	96.59	7.44
ResNet50	57.14	96.91	61.90	83.92	67.34	76.16
VGG19	75.10	60.10	87.75	32.46	88.44	37.76

4.1.2 Confusion Matrix. This matrix provides a comprehensive overview of the accuracy, precision, recall, and F1 score used by each model, which can be useful in determining the relative strengths and weaknesses of the models, particularly concerning unbalanced datasets.

Figure 7 illustrates the confusion matrix for the MobileNetV2, VGG19, InceptionV3, and ResNet50 which shows the performance in classifying four types of fruit conditions: blackspot, canker, greening, and healthy fruit.

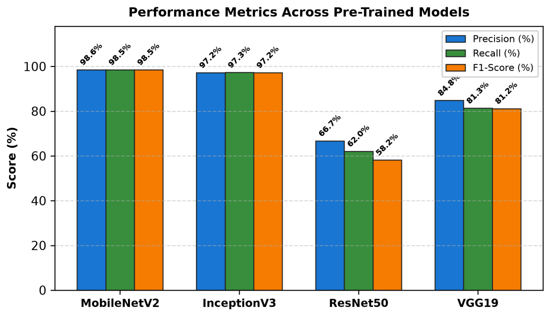

Fig. 8. Comparative diagnostic metrics across the four pre-trained architectures.

4.2 Complexity Analysis

The complexity analysis is an important factor in evaluating the models. Training time, inference time, and memory usage are evaluated for each model to compare their performance.

Table 8. Quantitative Computational Complexity Metrics Across Models

Metric	MobileNetV2	InceptionV3	ResNet50	VGG19
Training Time (s)	406.29	717.00	1040.00	3360.14
Inference Latency (s)	83.67	85.73	84.11	80.34
Memory Footprint (MB)	9.82	22.81	34.28	118.59

Table 8 compares four pre-trained models: MobileNetV2, InceptionV3, ResNet50, and VGG19 based on training time, inference time, and memory usage. MobileNetV2 is the most efficient with the shortest training time of 406.29 seconds and the lowest memory usage of 9.82 MB, while VGG19 has the highest training time of 3360.14 seconds and the highest memory usage of 118.59 MB. Inference times are relatively close across all models, with VGG19 slightly leading at 80.34 seconds.
Figure 8 shows that MobileNetV2 had the highest performance in the above metric with a precision of 98.56%, 98.54% for recall, and 98.54% in the F1 score hence highly efficient in the classification of the disease.

4.3 Comparative Discussion

This section will give a comparative discussion of used models during citrus fruit disease diagnosis using their performance and complexity metrics. In this work, the models of four pre-trained models, namely VGG19, ResNet50, InceptionV3, and MobileNetV2, have been evaluated on the citrus fruit diseases dataset, and the goal was to select the best model based on performance and complexity. The performance used to evaluate the models was precision, recall, F1-score, and accuracy.
MobileNetV2 and InceptionV3 have quite large training accuracies as well as low training loss, and seem to be tending toward generalization; in contrast, the accuracies of ResNet50 were significantly lower indicating sub-optimal learning and generalization properties.
After comparing all the analyses, it is found that MobileNetV2 showed the best performance in case of accuracy, memory usage, training time, and testing time. Also, the lightweight architecture makes it suitable to deploy on real-time applications.

5. CONCLUSION

This research work highlights the importance of transfer learning with four pre-trained models such as MobileNetV2, InceptionV3, ResNet50, and VGG19 while detecting citrus fruit diseases. The study concluded that MobileNetV2 is the best pre-trained model yielding high accuracy for identifying citrus fruit disease from other models (VGG19, ResNet50, and InceptionV3). It fetches the highest precision, recall, and F1 score with the dataset. Thus it has made an excellent training and testing accuracy as well. InceptionV3 closely finished behind the strong numbers but ResNet50 was far back. VGG19 resulted in a decent performance. In the end, MobileNetV2 wins with the highest value for the problem set, hitting both high accuracy and efficiency thus it is most effective in terms of performance and complexity.

5.1 Limitations

This study has several limitations. First, the proposed models were evaluated using only a single citrus fruit dataset consisting of 1,463 images, which is relatively small for training and evaluating deep learning models. Future research should therefore focus on expanding the dataset and integrating a broader range of citrus disease categories to develop more robust and comprehensive disease classification models.

5.2 Future Scope

The implementation of bio-inspired models, e.g. GWO (Grey Wolf Optimizer), Particle Swarm Optimizer (PSO), etc. can be a new direction in future work. In addition, building hybrid models merging the best of different architectures could be beneficial to enhance accuracy and robustness. This would be better solved by including explainable AI techniques, as these make it easier to comprehend the model's decision process and lead to trust and adoption in practice.

6. REFERENCES

[1] Ahmed Abdelmoamen Ahmed and Gopireddy Harshavardhan Reddy. A mobile-based system for detecting plant leaf diseases using deep learning. *AgriEngineering*, 3(3):478–493, July 2021.

[2] T. Ahmed, A. Muqit, J. Datta, M. Haque, and M. K. Haque. Prevalence and severity of different citrus diseases in sylhet region. *Journal of Bioscience and Agriculture Research*, 23(2):1957–1968, 2020.

[3] Al-Kindi Publisher. Al-kindi publisher: Open access journals and books, 2025. Accessed: 2 March 2025.

[4] Vasileios Balafas, Emmanouil Karantoumanis, Malamati Louta, and Nikolaos Ploskas. Machine learning and deep learning for plant disease classification and detection. *IEEE Access*, 11:114352–114377, 2023.

[5] Nouman Butt et al. Citrus disease detection and classification using machine learning techniques. *Journal of Computing & Biomedical Informatics*, 6(02), March 2024.

[6] Poonam Dhiman, Amandeep Kaur, Yasir Hamid, Eatedal Alabdulkreem, Hela Elmannai, and Nedal Ababneh. Smart disease detection system for citrus fruits using deep learning with edge computing. *Sustainability*, 15(5):4576, March 2023.

[7] Ebin.pub. Ebin.pub: Free ebooks and digital resources, 2025. Accessed: 2 March 2025.

[8] Ahmed Elaraby, Walid Hamdy, and Saad Alanazi. Classification of citrus diseases using optimization deep learning approach. *Computational Intelligence and Neuroscience*, 2022:1–10, February 2022.

[9] Frontiers. Frontiers: Peer-reviewed research articles. https://www.frontiersin.org, 2025. Accessed: 3 March 2025.

[10] Wilfrido Gómez-Flores, Juan José Garza-Saldaña, and Sóstenes Edmundo Varela-Fuentes. Citrusuat: A dataset of orange citrus sinensis leaves for abnormality detection using image analysis techniques. *Data in Brief*, 52:109908, February 2024.

[11] Tareq Hasan, Marjuk Ahmed Siddiki, and Md Naim Hossain. Detection of bangladeshi-produced plant disease using a transfer learning based on deep neural model. *Journal of Computer Science and Technology Studies*, 5(3):55–69, August 2023.

[12] Hindawi. Hindawi: Open access publisher, 2025. Accessed: 2 March 2025.

[13] Journal of Theoretical and Applied Information Technology. Official website of jatit. http://www.jatit.org, 2025. Accessed: 3 March 2025.

[14] Ejaz Khan, Muhammad Zia Ur Rehman, Fawad Ahmed, and Muhammad Attique Khan. Classification of diseases in citrus fruits using squeezenet. In *2021 International Conference on Applied and Engineering Mathematics (ICAEM)*, pages 67–72. IEEE, August 2021.

[15] Asad Khattak, Muhammad Usama Asghar, Ulfat Batool, Muhammad Zubair Asghar, Hayat Ullah, Mabrook Al-Rakhami, and Abdu Gumaei. Automatic detection of citrus fruit and leaves diseases using deep neural network model. *IEEE Access*, 9:112942–112954, 2021.

[16] Vinay Kukreja, Rishabh Sharma, and Rishika Yadav. Improving citrus farming through efficient and accurate diagnosis of lemon citrus canker disease with deep learning. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE, May 2023.

[17] Madhusudan G. Lanjewar and Jivan S. Parab. Cnn and transfer learning methods with augmentation for citrus leaf diseases detection using paas cloud on mobile. *Multimedia Tools and Applications*, 83(11):31733–31758, September 2023.

[18] Dehuan Luo, Yueju Xue, Xinru Deng, Bin Yang, Haifei Chen, and Zhujiang Mo. Citrus diseases and pests detection model based on self-attention yolov8. *IEEE Access*, 11:139872–139881, 2023.

[19] Nafisha Binte Moin, Nabila Islam, Shamima Sultana, Lubaba Alam Chhoa, S. M. Ruhul Kabir Howlader, and Shamim H. Ripon. Disease detection of bangladeshi crops using image processing and deep learning - a comparative analysis. In *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pages 1–8. IEEE, June 2022.

[20] Patrick Rexrode Murphy. *Self-Care and Quality of Life of Remotely Monitored Appalachians with Heart Failure*. PhD thesis, West Virginia University Libraries, 2023.

[21] Sana Parez, Naqqash Dilshad, Turki M. Alanazi, and Jong Weon Lee. Towards sustainable agricultural systems: A lightweight deep learning model for plant disease detection. *Computer Systems Science and Engineering*, 47(1):515–536, 2023.

[22] PeerJ. Peerj: Open access research journal. https://peerj.com, 2025. Accessed: 3 March 2025.

[23] Research Square. Research square: Accelerating the dissemination of research, 2025. Accessed: 2 March 2025.

[24] ResearchGate. Researchgate: Professional network for scientists, 2025. Accessed: 2 March 2025.

[25] Ashok Kumar Saini, Roheet Bhatnagar, and Devesh Kumar Srivastava. Sadcnn-orbm: a hybrid deep learning model based citrus disease detection and classification. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(2):2191, April 2024.

[26] Ambreen Shah, Muhammad Attique Khan, Ahmed Ibrahim Alzahrani, Nasser Alalwan, Ameer Hamza, Suresh Manic, Yudong Zhang, and Robertas Damaševičius. Fuzzyshallow: A framework of deep shallow neural networks and modified tree growth optimization for agriculture land cover and fruit disease recognition from remote sensing and digital imaging. *Measurement*, 237:115224, September 2024.

[27] Muhammad Sharif, Muhammad Attique Khan, Zahid Iqbal, Muhammad Faisal Azam, M. Ikram Ullah Lali, and Muhammad Younus Javed. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Computers and Electronics in Agriculture*, 150:220–234, July 2018.

[28] Josephin Shermila P, Akila Victor, S. Oswalt Manoj, and E. Anna Devi. Automatic detection and classification of disease in citrus fruit and leaves using a customized cnn based model. *Boletin Latinoamericano y del Caribe de Plantas Medicinales y Aromaticas*, 23(2):180–198, March 2024.

[29] Staffordshire University ePrints Repository. Staffordshire university eprints repository. https://eprints.staffs.ac.uk, 2025. Accessed: 3 March 2025.

[30] R. Sujatha, Jyotir Moy Chatterjee, NZ Jhanjhi, and Sarfraz Nawaz Brohi. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems*, 80:103615, February 2021.

[31] Tampere University Repository (Trepo). Trepo: Open institutional repository of tampere university. https://trepo.tuni.fi, 2025. Accessed: 3 March 2025.

[32] Uppsala University DiVA Portal. Diva portal: Academic archive of uppsala university. https://uu.diva-portal.org, 2025. Accessed: 3 March 2025.