GreenGuard: An Android-Based Application for Early Disease Detection of Indoor Plant

Md. Mahmudur Rahman[1] and Samira Akter Tumpa [1]

[1] Institute of Information Technology, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh
mahmud@juniv.edu

Abstract. Numerous individuals like the hobby of taking care of indoor plants, which provides both aesthetic and health benefits. However, they may be susceptible to a variety of disorders, as is the case with other plants. Manually identifying plant diseases is a time-consuming, error-prone, and unreliable process that impedes the ability to effectively identify and prevent them. The challenges can be surmounted by the early and precise detection of interior plant diseases by implementing new technologies such as Deep Learning and Machine Learning . In this research, we introduce 'GreenGuard,' a revolutionary solution for indoor plant devotees. This study employs critical deep-learning models to identify plant diseases, including ResNet50, VGG16, EfficientNetB0, EfficientNetB5, and EfficientNetB7. Models such as ResNet50 and EfficientNetB7 are more effective in detecting maladies in indoor plants regarding accuracy, precision, recall, and F1 score. Our Android application enables the early detection of common diseases that affect indoor plants, particularly those on fashionable balconies, and provides solutions by utilizing ResNet50's advanced image processing techniques.

Keywords: Indoor Plant Disease Detection, Deep Learning, ResNet50, Android App, Image Processing.

1 Introduction

As urbanites come to terms with living in high-rise apartments and squeeze every bit of life out of the 24-hour day, indoor plants have become an increasingly popular home decor trend. These plants act as adornments and also support the mental health by connecting to nature in urban settings [1]. But keeping indoor plants alive is much harder, especially for urbanites with limited gardening experience. Such fungal infections which are common as well as infestations by ants, whiteflies, mealybugs and viral diseases many of them completely unnoticed until they damaged the health plants [2]. Most of the traditional disease detection methods [3] are less effective and they do not provide immediate action required for further intervention. GreenGuard utilizes cutting-edge image processing technology to capture images of their indoor plants, which can accurately identify plant diseases, and thus assisting plant enthusiasts in preventing hazardous situations. Users can easily open the mobile application, access the camera option, capture an image of the disease, and let the application identify the disease and suggest preventive measures using digital image processing technology. We've incorporated deep learning models and an android application to simplify our tasks. Despite the existence of various plant disease detection technologies, our research distinguishes itself by concentrating on identifying various diseases in indoor plants, which serve as decorative elements and enhance our mental well-being [4]. This integration of technology and horticulture aims to empower users with the knowledge and tools necessary to maintain their indoor plants' vitality and longevity. Through GreenGuard, we strive to transform urban residents' care for their indoor plants, fostering healthier and more vibrant green spaces within their homes.

In recent years, the increasing popularity of indoor plants as elements of interior design and well-being has brought attention to the challenges associated with maintaining their health. Traditional disease detection methods often lack efficiency and real-time feedback [5]. When searching online, people who are dealing with infections in their houseplants frequently fail to locate the right remedy. When a fungus attacks an indoor plant, the majority of users are unaware of the proper medication to use. Ants, fungus, mealybugs, and whiteflies can sometimes bring plants to the brink of death. Our research stems from the desire to address these limitations by implementing the most recent technology. We have created a mobile application that empowers balcony gardeners to effortlessly identify and avert common indoor plant diseases like fungi, ants, whiteflies, mealybugs, and viruses. The study also aims to compare various deep-learning methods to enhance the accuracy of disease identification and provide effective solutions for urban residents. The visual mobile application identifies potential health issues and offers users actionable insights and solutions, contributing to the overall well-being of indoor green spaces. The research contributes in the following ways:

- A user-friendly solution for indoor plant disease detection and preventive measures among urban residents is developed.
- A mobile application is developed that can detect and prevent indoor plant diseases like fungus, ants, whiteflies, mealybugs, and other viruses, thereby reducing harm to indoor plants.
- Deep learning models such as ResNet50, VGG16, EfficientNetB0, EfficientNetB5, and EfficientNetB7 are compared and implemented for accurate plant disease identification.

2 Related Work

Mukti et al. [7] utilized transfer learning with the ResNet50 architecture model that achieved an impressive training accuracy of 99.80% in identifying plant diseases and showcasing the power of deep learning in detecting plant diseases. Their dataset consists of 70295 training images and 17572 validation images containing 38 different plant leaf classes. Arshad et al. [8] delve into the use of deep learning techniques, particularly ResNet50 with Transfer Learning, to detect diseases in potatoes, tomatoes, and corn. With ResNet50, they were able to achieve an impressive accuracy of 98.7%, which ultimately led to improved crop yield. Thakur et al. [**9**], utilized three different CNN architectures - InceptionV3, ResNet50, and VGG16, with transfer learning to identify bacterial spot disease in Bell Pepper plants. The results were quite impressive, with accuracies reaching as high as 99.72% for VGG16, 99.31% for ResNet50, and 95.77% for InceptionV3. Their work greatly enhances the detection of diseases in agriculture, facilitating prompt interventions. Chellapandi [**10**] investigates deep learning and transfer learning models for classifying images of diseased plant leaves. Their research showcases DenseNet as the top-performing model, with an impressive 99% accuracy on test data. This underscores its potential in detecting crop diseases at an early stage. Research conducted by Pradhan et al. [**11**] delves into the use of deep learning techniques to accurately identify apple diseases from digital images. This study showcases the superior performance of DenseNet201 compared to other models, achieving an impressive accuracy rate of 98.75%. It emphasizes the effectiveness of CNN-based methods in the field of plant disease management. Sanida et al. [**12**] devised a two-stage transfer learning method for identifying tomato leaf diseases. They utilized a unique network based on VGGNet with two inception blocks. With their expertise in data analysis, their model achieved an impressive 99.23% accuracy. This breakthrough will greatly benefit farmers by improving tomato crop management and disease detection.

In [13], Li et al. go into excellent detail about how deep learning has helped find crop leaf diseases. The review talks about recent progress, ongoing problems, and the need for more advanced imaging methods. It gives experts useful information for finding plant diseases and controlling insect pests. In the real world, Arsenovic et al. [14] created a new set of 79,265 labeled pictures of leaves. They suggested new ways to improve existing plant disease detection models by using generative adversarial networks (GANs) and a two-stage neural network design. These methods worked amazingly well, achieving a 93.67% success rate. Ale et al. [15] describe a DenseNet-based transfer learning method for finding diseases on edge servers and a simple DNN method for Internet of Things (IoT) devices. Their models show how to accurately find diseases while using as few computing resources as possible, which is important for smart farm systems to work well. They [16] compared hyperparameters in deep learning models by analyzing 160 papers from research on plant disease detection, discussing the relevance of their findings.

Ramesh et al. [17] introduced a process with Random Forest and Histogram of Oriented Gradient (HOG) to correctly identify infested leaves versus healthy ones, showing the power of machine learning in creating massive plant disease detections. Varshney et al. [18] literature proposed a new methodology for the detection and classification of leaf plant disease applying transfer learning with CNN as a feature extractor and SVM as a classifier. The training accuracy of their model exceeded the one reported for other molecular biology and image processing techniques, providing a significant step toward binary plant disease identification in agriculture. In these two literatures, one by [19] reported the classification of plant leaf diseases using KNN, resulting in an accuracy of 98.56%, whereas another article by [20] illustrates how machine learning techniques assist in predicting crop diseases accurately through feature reduction, leading to dimensionality reduction that improves food security across the world.

Advances in hyperspectral technology for precision agriculture are improving plant disease diagnosis and this study [21] emphasizes the advantages of hyperspectral imaging compared with visual observation in disease recognition. It suggests novel strategies for identifying pathogens and an early warning system. Kuswidiyanto et al. [22] discuss advances in disease detection methods and examined how hyperspectral remote sensing systems on UAVs may be integrated with deep learning algorithms to identify plant diseases effectively at a low cost. Terentev et al. [23] has provided an in-depth review of hyperspectral remote sensing for early plant disease detection, with emphasis on the prospective methodological developments and how semantically rich data from Hyperspectral sensors are important to determine crop health. Cheshkova [24] studying hyperspectral techniques that should be employed to identify diseased plant (healthy or sick). Early disease detection is especially crucial regarding quantification. Nevertheless, technological barriers remain to implementing these advances in a real-world setting. More analysis is needed for this.

Limitations in previous research on plant disease detection suggest the importance of creating tailored techniques specifically for indoor settings. While most research focuses on a wider agricultural context, some of it is not directly applicable to the more unique indoor constraints found in sheltered locations (e.g. balconies). However, the lack of self-collected datasets which are significantly concentrated on indoor foliage act as a barrier in increase accurate detection algorithms. Most of the methods that work in larger agricultural settings do not translate well into many indoor plant maintenance applications. Furthermore, the scarcity of indoor plant disease detection applications hints an area that ought to be addressed in order for disease management efforts to succeed.

3 Methodology

3.1 Proposed Method

During last few years, it is noticeable that deep learning technologies have been widely used in the area of plant disease detection also applicable within indoor. This thesis investigates the methodology using Convolutional Neural Networks (CNNs), specifically ResNet50 and VGG16, to identify indoor plant diseases. CNNs are a type of neural network used to learn spatial hierarchies of features from raw sensory data, they automatically and adaptively perform feature extraction in the context of visual information which may be particularly well-suited for image-based tasks like plant disease classification. ResNet50, a variant of CNNs, which uses residual learning to solve vanishing gradient problem and helps in training of deeper networks with better accuracy. In contrast, VGG16 adopts a simpler architecture with smaller filter sizes and deeper layers resulting in better performance on image recognition. The present study expects to contribute to improving the indoor plant disease detection accuracy and efficiency by using the impressive capabilities of these deep learning models, for promoting precision agriculture delivering sustainable crop management practices. The work in this research is aimed at creating a plant disease diagnostic application based on image classification using various deep learning models.

The chosen framework, a user-friendly web tool in **Fig. 1**. facilitates deep learning models' fast and accurate creation.

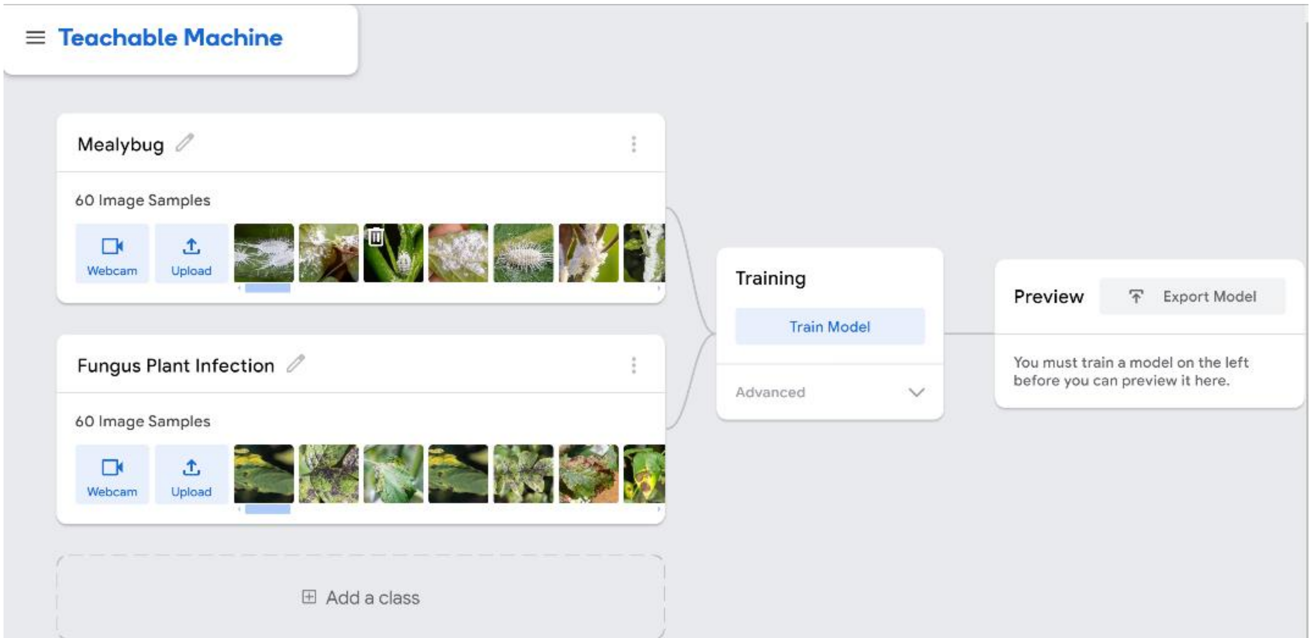


Fig. 1. General View of Web-Based Tool

Here, from creating a new image project, we develop an Image Classification application that specifically focuses on eight common indoor plants diseases [25]. Different varieties of plant images are uploaded using the default image module after which the model goes through training and finally, we export our model as a machine learning TensorFlow model and finally load to android studio without any discrepancy for deployment [26]. This brief workflow in **Fig. 2**. describes the workflow followed to design and deploy a plant disease classification system.

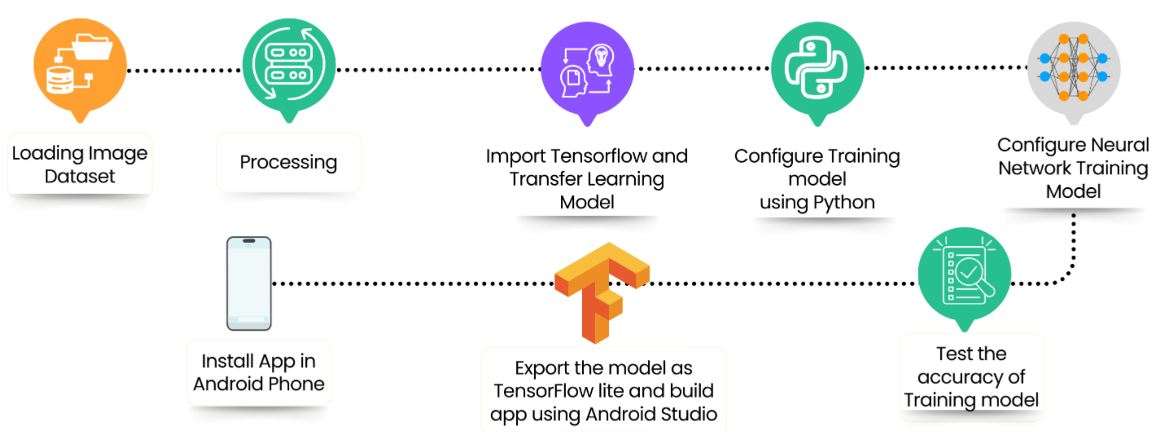


Fig. 2. Proposed workflow of the disease detection of indoor plants

3.2 Data Collection and Summary of Dataset

We conducted a thorough in-vitro assessment at plant nurseries in the Savar area to detect indoor plant diseases. The first aim was to find out about the diseases people usually have on their indoor plants, and how these differ from diseases that affect outdoor trees. These nurseries provided invaluable perspectives on the challenges that diseases pose to indoor plants through long discussions with experts. Moreover, the fieldwork yielded substantial image data that covered various stages and forms of these diseases. This is how, the aforementioned curated dataset was used for training our machine learning model, resulting in a robust and accurate system to detect indoor plant diseases. The integration of real-world observations and image data from plant nurseries in Savar serves as a foundation for the actual findings presented in this thesis, enhancing the credibility and relevance of our research.

For our machine's training, there were about sixty images of Mealybugs, ants attacking plants, Fungus, white flies, Powdery miildew, leaf spot, gray mold and sixty images of root rot. **Table 1**. shows the summarized dataset.

Table 1. Summarizing the dataset of plant disease.

Sl	Plant Disease	Name Number of Images
1.	Mealybug	60
2.	Ant	60
3.	Fungus	60
4.	White fly	60
5.	Powdery mildew	60
6.	Leaf spot	60
7.	Gray mold	60
8.	Root rot	60

Home plant diseases from **Fig. 3**. shows important differences between indoor and outdoor plant diseases. One reason is the environmental conditions in which a plant grows-indoor plants are typically protected from weather changes, while outdoor variables include fluctuating temperature and humidity levels with year-round exposure to rain. Even assuming pathogen types are secondarily different (indoor plants may be quite prone to fungal infections in stagnant and humid air, while outdoor ones face a much larger suite of pathogens including bacteria, viruses and pests [27]). Thirdly, it is managed and controlled differently; indoor plant diseases usually need exacting watering/ventilation/cleansiness to keep them from spreading whereas outdoor ones may require integrated pest management (IPM), protective measures such as fungicides or insecticides [28]. In general, these distinctions are important for the appropriate diagnosis and treatment of plant disease in both enclosed or outdoor environments.

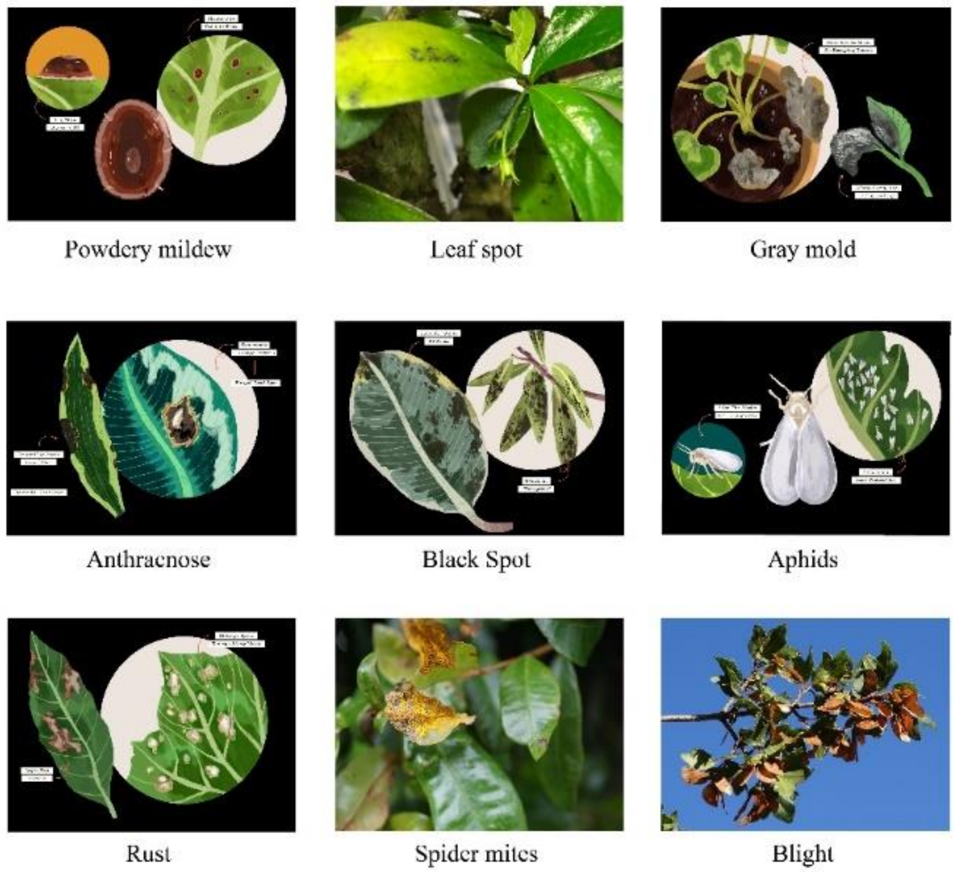


Fig. 3. Sample of Indoor plant's disease

3.3 Data Preprocessing

We collected data from a few plant nurseries and photographed the plants with a camera. We just borrowed a few images from the internet. Considering the noise in it, the brightness and saturation need to be perfect as well. For image augmentation, the images have to be all resized. So the process consist of two steps:

- Image scaling
- Augmentation

3.3.1 Image scaling

Fundamentally image scaling or resizing is a preliminary stage of image processing in plant disease detection. The adjustments of the dimensions to one image size compatibility within dataset is a part of it. When we regularize the size of plant images it ensures that no deformation gets introduced and makes dataset more harmonized to analyze successfully. This step helps us throughout our research as providing with the same dataset that means we will have accurate and similar training accuracy for machine learning models of plant disease detection.

3.3.2 Augmentation

Image augmentation technique used in some of the fundamental tasks which help to increase the number and quality of data available for training machine learning model. This technique has a number of methods to augment input images such as rotation, scaling, flip the image and changing colors at different saturation levels in order to create new synthetic data. Augmented images in **Fig. 4**. further enhance model robustness and generalization, as the model is exposed to a more diverse range of cases or variations [29]. The key application area of image augmentation in plant disease detection through an android app is to increase the dataset with different lighting conditions, angles and severity levels of diseases which will help the model for better identification and classification among all possible types of plant diseases more accurately.

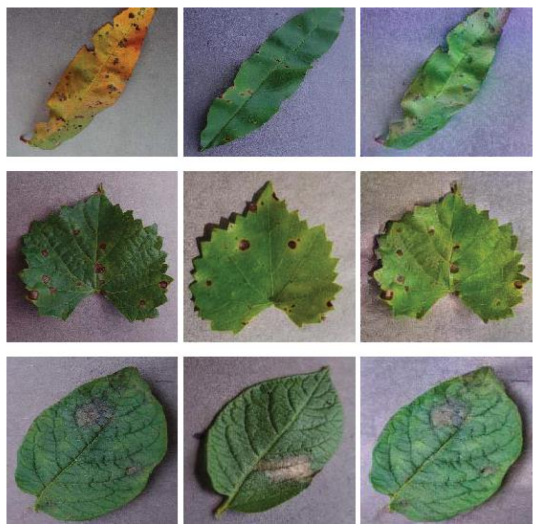


Fig. 4. Image Augmentation

3.4 Classification Method

In general, this classification approach based on the combination of CNNs, image processing, and deep learning to detect plant diseases can be considered a multi-method. CNN is a methodology that can automatically learn features from raw image data. Employing image processing techniques such as pre-processing, segmentation, and feature extraction enhances the quality of the input data for the CNN model based on [30]. Deep learning algorithms, including ResNet50 (a CNN architecture), VGG16, EfficientNet, and others, utilize the processed image data to train and learn intricate patterns related to various plant diseases. The classification process involves labeling or categorizing each input image based on the diseases it contains, and models that rely on this process have already learned how to represent these diseases through decision-making [31]. Such holistic functionality enables accurate disease identification and lays the groundwork for complete automation in efficient plant health monitoring systems.

3.4.1 ResNet50 (Residual Networks)

Microsoft Research created the powerful deep-learning model ResNet50 in 2015. They performed pretraining on ImageNet using a 50-layer residual network [32]. The ResNet50 Model is a Convolutional Neural Network (CNN) that has already undergone training for image categorization. It is built upon a residual network architecture, a design specifically designed to address vanishing gradients in deep neural networks. Since then, the model has become more effective as it can learn more complex features from the network. This architecture's ability in **Fig. 6**. to maintain a more accurate and efficient model while allowing for a deeper level of network layers is one of its primary features. ResNet50 is frequently used in computer vision applications like object detection, image segmentation, and image recognition since it has produced state-of-the-art outcomes in image classification tasks. One of its key features is the ability to have a more accurate and efficient model while still benefiting from longer network layers, as shown in **Fig. 6**. Visual applications like object detection, image segmentation, and face recognition often utilize ResNet50 due to its state-of-the-art accuracy on tasks like imagenet classification.

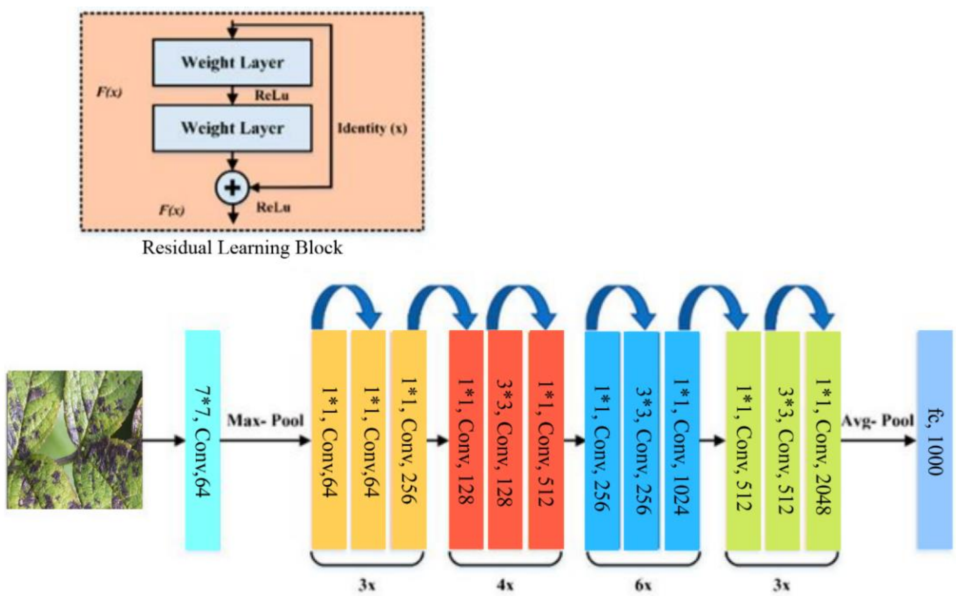


Fig. 6. ResNet50 Architecture [32]

3.4.2 VGG 16 (Visual Geometry Group)

The Visual Geometry Group, University of Oxford, proposed VGG16, which is renowned for its ease of understanding image classification tasks. It is a 16-layer network that just contains only convolutional layers as used at the beginning of other networks and nothing else (i.e., no fully connected layer or softmax, which are usually in CNNs). The secret to reaching really deep nets is the use of small (3x3) receptive fields throughout the network. VGG16 received such high attention because it was very successful in the 2014 ImageNet

Large Scale Visual Recognition Challenge (ILSVRC), where it performed top results for object detection and classification.

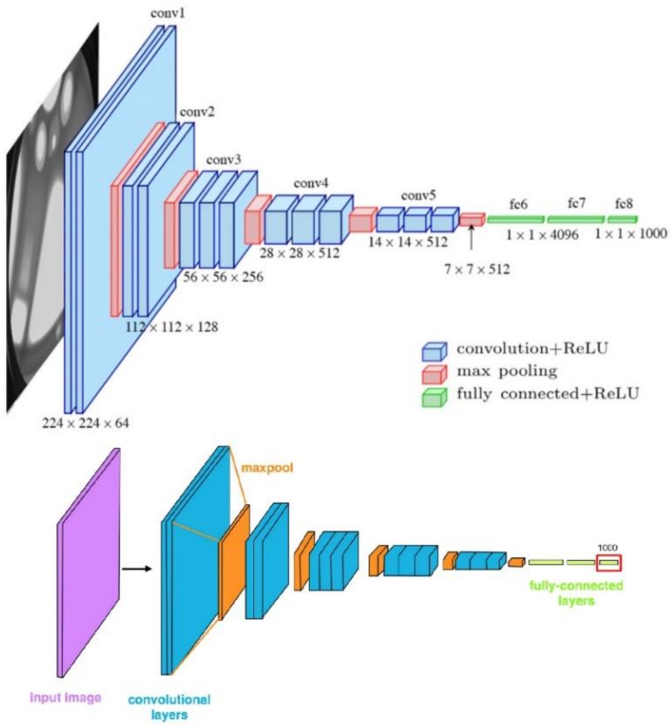


Figure 6: VGG 16 Architecture [33]

3.4.3 EfficientNet

EfficientNet [34] is a family of convolutional neural networks (CNNs) developed by Google AI, designed to achieve high accuracy while maintaining efficiency in terms of computational resources. The EfficientNet models, ranging from B0 to B7, utilize a compound scaling method that uniformly scales the network's depth, width, and resolution, optimizing the balance between accuracy and efficiency. EfficientNet-B0 serves as the baseline model, and subsequent models (B1 to B7) are scaled versions of B0, each progressively larger and more powerful. These models have been shown to outperform many existing CNNs in terms of both accuracy and computational efficiency, making them highly suitable for a wide range of image classification tasks.

Table 02. Comparison of EfficientNetB0, B5, and B7.

Feature	EfficientNetB0	EfficientNetB5	EfficientNetB7
Parameters	5.3 million	30 million	66 million
FLOPS	0.39 billion	9.9 billion	19 billion
Input Resolution	224 x 224	456 x 456	600 x 600
Top-1 Accuracy	77.1% on ImageNet	83.6% on ImageNet	84.4% on ImageNet
Efficiency	Baseline model	More computationally intensive	Most computationally intensive

3.5 Experimental Discussion

This experiment uses Python and Jupyter Notebook version 6.4.12 for all work. We used 70% of the data from the set for training, 10% of the data for testing, and 20% of the data for validation. To generate and identify different classification architectures, we employed the EfficientNetB0, EfficientNetB5, EfficientNetB7, ResNet50, and VGG16 methods. After that, check the final model summary, compilation, and fitting. We implemented all these classifiers using the Scikit-Learn library. Finally, we generated a classification report. This report provides information on accuracy, accuracy loss, precision, recall, and the F1-score.

3.5.1 Evaluation Metrics

In order to evaluate the performance of different classifiers, a number of evaluation metrics like Accuracy, F1-Score, Precision, and Recall are used. However, these metrics are manipulated using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values which are described as follows:

Accuracy: is calculated as the ratio of the total population to the sum of the TP and TN.

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

F1-Score: accumulates the precision and recall and manipulates the harmonic mean of them.

$$F - Measure = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision + recall}} = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \qquad \text{(ii)}$$

Precision: This is determined by dividing the total number of true positives and false positives by the imbalanced classification problem's two classes.

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

Recall: is just the type II error rate's complement, or one minus the type II error rate.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad \text{(iv)}$$

3.6 Prototyping Method

This research have used prototyping process. **Fig. 7.** gives overall idea of the system development cycle with a close look at object oriented approach in System analysis and prototype technique. This project uses teachable machines for machine learning model [35]. This allowed to be trained and run supervised models with TensorFlow.js, a Javascript machine learning toolkit on a platform called Teachable Machine developed by Google. These models employ a method known as transfer learning, which explains most of the architecture of neural networks. MobileNet is a pre-trained image recognition network that powers the Google Teachable Machine.

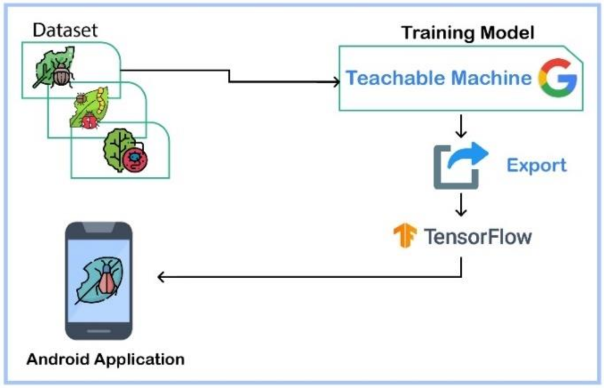


Fig. 7. Prototyping for Android Application

4 Experimental Evaluation

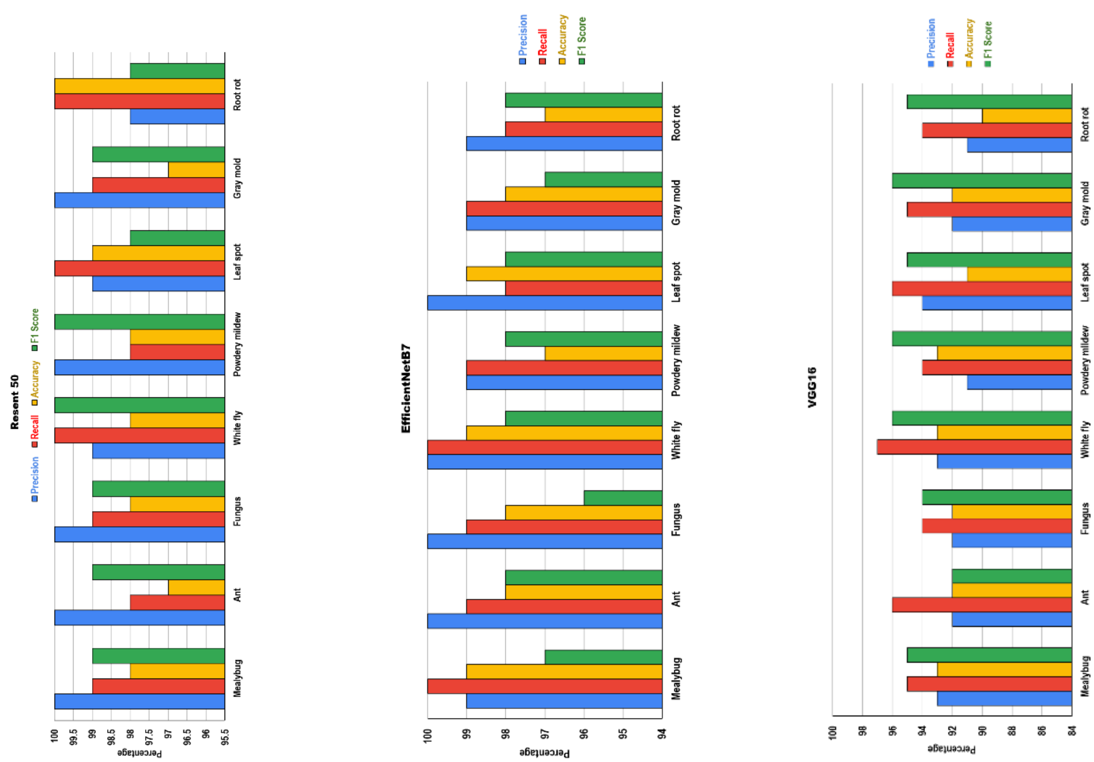
4.1 Performance of Classifiers

The experimental results of different classification algorithms are shown in **Table 03**. To support our findings, we used a variety of evaluation metrics, including Accuracy, Precision, Recall, and F1-Score. All classifier results were more than 80%. The results from ResNet50 are the most rewarding with accuracy of 98.06%, Recall of 99.12%, F1-Score of 98.81% and Precision of 99.5%. EfficientNetB7 has also performed well and is almost comparable with ResNet50 as a classification algorithm. On the other hand, EfficientNetB0 performs poorly, with Accuracy being only 78.5%, Recall being only 81.5%, F1-Score being only 81.12%, and precision being only 80.38%. Other classification methods have also produced average results. In VGG16, we have an Accuracy of 92.25%, Recall of 95.12%, F1-Score of 94.625%, Precision of 92.25%. Additionally, EfficientNetB5 is quite equal to the VGG16, where we got an Accuracy of 92%, Recall of 95.12%, F1-Score of 94.625% and Precision of 94%.

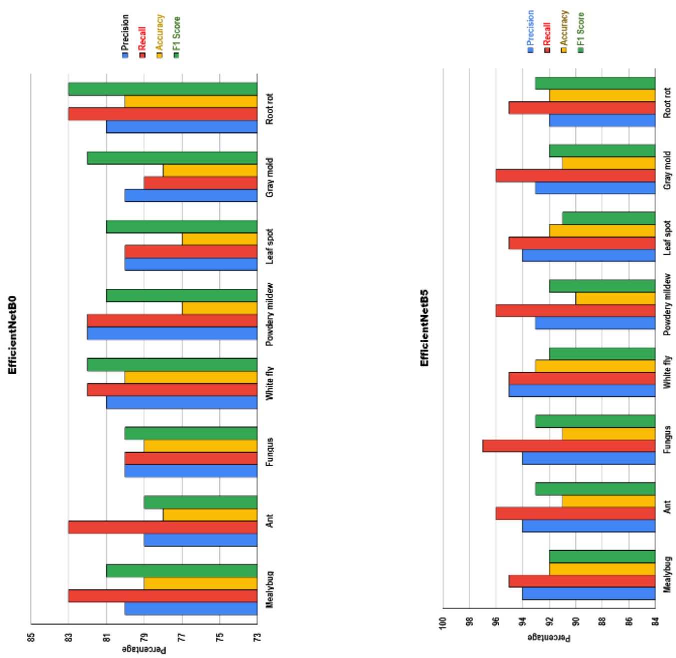
Table 3. Average Performance of Individual Model

Model	Average Precision (%)	Average Recall (%)	Average Accuracy (%)	Average F1 Score (%)
ResNet50	**99.5**	**99.125**	98.0625	**98.8125**
EfficientNetB7	99.5	99	**98.125**	97.5
EfficientNetB5	94	95.125	91.5	94.625
VGG16	92.25	95.125	92	94.625
EfficientNetB0	80.375	81.5	78.5	81.125

For robust performance among the above mentioned model, we have applied ResNet50 for plant disease classification and disease detection. We gained all the classifier's results that were more than 90%. For Mealybug, the evaluation metrics, including accuracy, accuracy loss, precision, recall, and F1-score, are respectively more than 98%. The evaluation metrics, including accuracy, accuracy loss, precision, recall, and F1-score, are respectively more than 96%. For fungus, the evaluation metrics, including accuracy, accuracy loss, precision, recall, and F1-score, are respectively more than 98%. For white fly, the evaluation metrics, including accuracy, accuracy loss, precision, recall, and F1-score, are respectively more than 98%. For powdery mildew, the evaluation metrics, including accuracy, accuracy loss, precision, recall, and F1-score, are respectively more than 98%. Moreover, we have analyzed every classification method and its results. We will better understand by focusing on the following figures. Secondly, we have gradually applied VGG16, EfficientNetB0, EfficientNetB5, and EfficientNetB7. Finally, we have gained the best result from the classification method ResNet50. Now we will implement our best model for Android application. We can see the result in the **Fig.** 8. showcased below.



(a) (b) (c)



(d) (e)

Fig. 8. Performance of (a) ResNet50 (b)EfficientNetB7 (c)VGG16 (d) EfficientNetB0 (e) EfficientNetB5

4.2 Individual Performance Result

A detailed comparison of various machine learning models (ResNet50, VGG16, EfficientNetB0, EfficientNetB5, and EfficientNetB7) is shown in **Table 4**. in terms of their performance metrics—Precision, Recall, Accuracy, and F1 Score—across different plant diseases.

Table 4. Detailed Comparison of Different Models

Plant Disease	Model	Precision(%)	Recall (%)	Accuracy (%)	F1 Score (%)
Mealybug	ResNet50	100	99	98	99
	VGG16	93	95	93	93
	EfficientNetB0	80	83	79	81
	EfficientNetB5	94	95	92	92
	EfficientNetB7	99	100	99	97
Ant	ResNet50	100	98	97	98.5
	VGG16	92	96	92	92
	EfficientNetB0	79	83	78	79
	EfficientNetB5	94	96	91	93
	EfficientNetB7	100	99	98	98
Fungus	ResNet50	100	99	98	99

	VGG16	92	94	92	94
	EfficientNetB0	80	80	79	80
	EfficientNetB5	94	97	91	93
	EfficientNetB7	100	99	98	96
White fly	ResNet50	99	100	98	99.5
	VGG16	93	97	93	96
	EfficientNetB0	81	82	80	82
	EfficientNetB5	98	95	93	92
	EfficientNetB7	100	100	99	98
Powdery mildew	ResNet50	100	98	98	100
	VGG16	91	94	93	96
	EfficientNetB0	82	82	77	81
	EfficientNetB5	93	96	90	92
	EfficientNetB7	99	99	97	98
Leaf spot	ResNet50	99	100	98.5	98
	VGG16	94	96	91	95
	EfficientNetB0	80	80	77	81
	EfficientNetB5	94	95	92	91
	EfficientNetB7	100	98	99	98
Gray mold	ResNet50	100	99	97	98.5
	VGG16	92	95	92	96
	EfficientNetB0	80	79	78	82
	EfficientNetB5	93	96	91	92
	EfficientNetB7	99	99	97	97
Root rot	ResNet50	98	100	100	98
	VGG16	91	94	90	95
	EfficientNetB0	81	83	80	83
	EfficientNetB5	92	95	92	93
	EfficientNetB7	99	98	97	98

ResNet50 and EfficientNetB7 consistently attain the highest precision across the majority of plant diseases, with an average precision of 99.5%. This suggests a high degree of proficiency in accurately identifying affirmative cases. EfficientNetB7 and ResNet50 also exhibit strong recall performance in **Fig. 9**. with averages of 99% and 99.125%, respectively, indicating that they are capable of capturing all pertinent cases. EfficientNetB7 has an average accuracy of 98.125%, while ResNet50 is closely behind at 98.0625%. This is indicative of their overall efficacy in accurately classifying both positive and negative cases. ResNet50 obtains the highest average F1 Score of 98.8125%, which suggests a balanced performance in precision and recall.

Journal of Information Technology (2024):13, pp 1-20
ISSN: 2304-3237 (Online), 2227-1279 (Print)

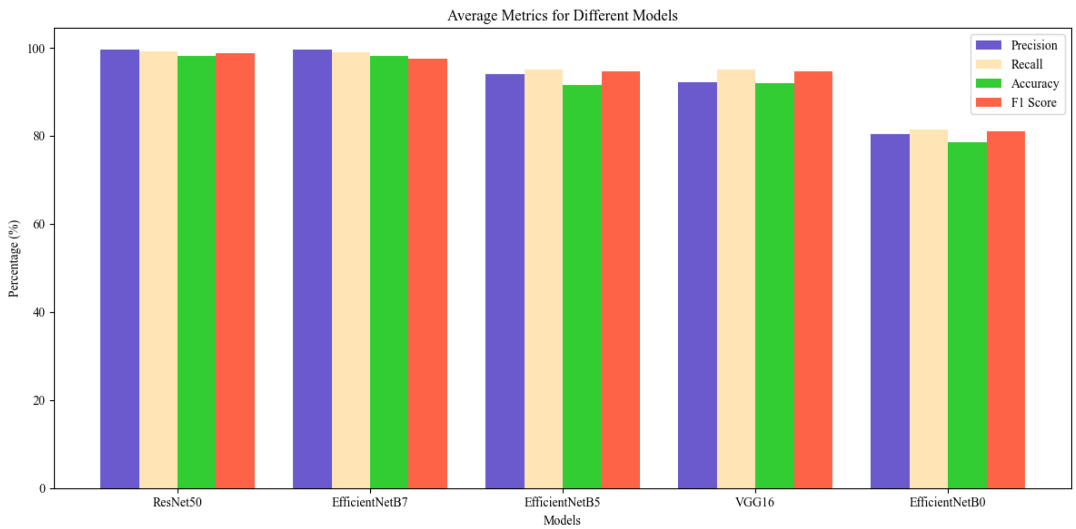


Fig. 9. Average Metrics for Different Models

From this, we can understand that ResNet50 and EfficientNetB7 are the most effective models in all metrics, rendering them suitable for plant disease classification tasks. They exhibit high precision, recall, accuracy, and F1 scores, which indicate their reliability and robustness in practical applications.

4.3 Mobile applications

Implementing ResNet50 in an Android application for plant disease detection involves several key steps. First, the pre-trained ResNet50 model must be adapted for mobile deployment, typically by converting it into a format suitable for inference on resource-constrained devices, such as TensorFlow Lite. This ensures efficient execution on Android platforms while maintaining high classification accuracy. Next, the Android application's user interface **Fig.10.** should be designed to allow users to capture images of plants using the device's camera. The captured images are then processed by the ResNet50 model to detect and classify shown in **Fig.11**. for any potential diseases present. Integration with digital image processing techniques may be necessary to preprocess the images and enhance their quality before inputting them into the model for classification.

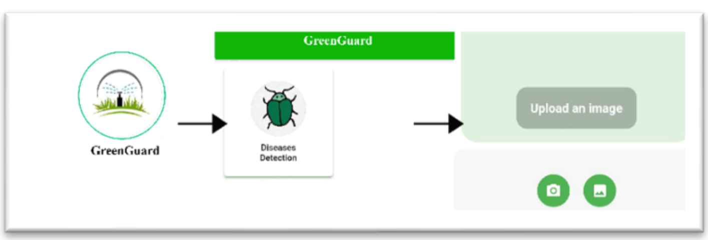


Fig. 10. GreenGuard interface

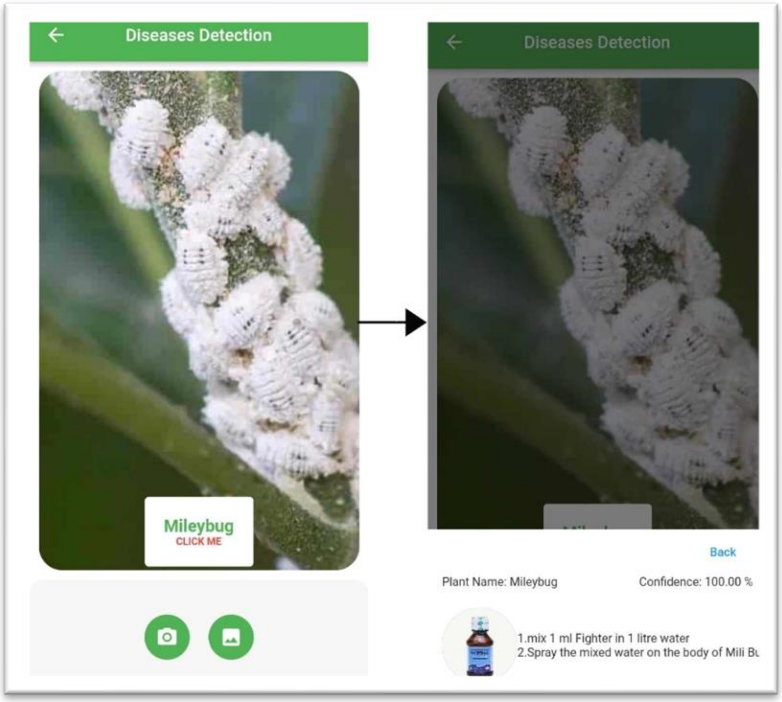


Fig.11. Plant Disease Detection Interface

5 Conclusion

To sum up, the formulation of the mobile application aimed to define and solve the illnesses in indoor plants should be a successful and advantageous solution for urban citizens who are interested in ornamental plants. The discussion has indicated that it is critical to focus on the problems that balcony gardeners may face, such as fungus, ants, whiteflies, mealybugs, and viruses. The findings may be useful for tailoring a user-friendly system to detect and prevent diseases in low-maintenance indoor plants. Therefore, the creation of Andrioid applications is beneficial for urban plant lovers as it meets their needs, and it is advantageous for indoor plants as it contributes to their wellbeing.

Future work plans may include further research focusing on the development and implementation of the advanced mobile application, which may combine image recognition technology and real-time data analysis. The inclusion of a community-sharing feature would also help to develop a more accurate and useful system. The future work could also explore partnerships with local nurseries or experts to provide customized recommendations for disease prevention and plant care. Additionally, considering the environmental impact, future developments may include eco-friendly solutions for insect control and disease management. Regular updates and user feedback mechanisms can further improve the app's effectiveness, ensuring continuous adaptation to face challenges in indoor plant care.

References

1 Sarkar, C., Gupta, D., Gupta, U., & Hazarika, B. B. (2023). Leaf disease detection using machine learning and deep learning: Review and challenges. Applied Soft Computing, 145, 110534.

2 J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, “Deep learning- based leaf disease detection in crops using images for agricultural applications,” Agronomy, vol. 12, no. 10, p. 2395, 2022.

3 M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, “Toled: Tomato leaf disease detection using convolution neural network,” Procedia Computer Science, vol. 167, pp. 293–301, 2020.

4 M. Türkoğlu and D. Hanbay, “Plant disease and pest detection using deep learning-based features,” Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, no. 3, pp. 1636–1651, 2019.

5 S. A. Burhan, S. Minhas, A. Tariq, and M. N. Hassan, “Comparative study of deep learning algorithms for disease and pest detection in rice crops,” in 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–5, IEEE, 2020.

6 M. V. Appalanaidu and G. Kumaravelan, “Plant leaf disease detection and classification using machine learning approaches: a review,” Innovations in computer science and engineering: Proceedings of 8th ICICSE, pp. 515–525, 2021.

7 Mukti, Ishrat Zahan, and Dipayan Biswas. "Transfer learning based plant diseases detection using ResNet50." 2019 4th International conference on electrical information and communication technology (EICT). IEEE, 2019.

8 M. S. Arshad, U. A. Rehman, and M. M. Fraz, “Plant disease identification using transfer learning,” in 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), pp. 1–5, IEEE, 2021.

9 P. Thakur, A. Chug, and A. P. Singh, “Plant disease detection of bell pepper plant using transfer learning over different models,” in 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 384–389, IEEE, 2021.

10 B. Chellapandi, M. Vijayalakshmi, and S. Chopra, “Comparison of pre-trained models using transfer learning for detecting plant disease,” in 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS),pp. 383–387, IEEE, 2021.

11 P. Pradhan, B. Kumar, and S. Mohan, “Comparison of various deep convolutional neural network models to discriminate apple leaf diseases using transfer learning,” Journal of Plant Diseases and Protection, vol. 129, no. 6, pp. 1461–1473, 2022.

12 T. Sanida, A. Sideris, M. V. Sanida, and M. Dasygenis, “Tomato leaf disease identification via two–stage transfer learning approach,” Smart Agricultural Technology, vol. 5, p. 100275, 2023.
13 L. Li, S. Zhang, and B. Wang, “Plant disease detection and classification by deep learning—a review,” IEEE Access, vol. 9, pp. 56683–56698, 2021.
14 M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, “Solv- ing current limitations of deep learning based approaches for plant disease detection,” Symmetry, vol. 11, no. 7, p. 939, 2019.
15 L. Ale, A. Sheta, L. Li, Y. Wang, and N. Zhang, “Deep learning based plant disease detection for smart agriculture,” in 2019 IEEE Globecom Workshops (GC Wkshps), pp. 1–6, IEEE, 2019.
16 C. Sunil, C. Jaidhar, and N. Patil, “Systematic study on deep learning-based plant disease detection or classification,” Artificial Intelligence Review, vol. 56, no. 12, pp. 14955–15052, 2023.
17 S. Ramesh, R. Hebbar, M. Niveditha, R. Pooja, N. Shashank, P. Vinod, et al., “Plant disease detection using machine learning,” in 2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C), pp. 41–45, IEEE, 2018.
18 D. Varshney, B. Babukhanwala, J. Khan, D. Saxena, and A. K. Singh, “Plant disease detection using machine learning techniques,” in 2022 3rd International Conference for Emerging Technology (INCET), pp. 1–5, IEEE, 2022.
19 A. S. Tulshan and N. Raul, “Plant leaf disease detection using machine learning,” in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6, IEEE, 2019.
20 A. Lowe, N. Harrison, and A. P. French, “Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress,” Plant methods, vol. 13, no. 1, p. 80, 2017.
21 N. Zhang, G. Yang, Y. Pan, X. Yang, L. Chen, and C. Zhao, “A review of advanced technologies and development for hyperspectral-based plant disease detection in the past three decades,” Remote Sensing, vol. 12, no. 19, p. 3188, 2020.
22 L. W. Kuswidiyanto, H.-H. Noh, and X. Han, “Plant disease diagnosis using deep learning based on aerial hyperspectral images: A review,” Remote Sensing, vol. 14, no. 23, p. 6031, 2022.
23 A. Terentev, V. Dolzhenko, A. Fedotov, and D. Eremenko, “Current state of hy- perspectral remote sensing for early plant disease detection: A review,” Sensors, vol. 22, no. 3, p. 757, 2022.
24 A. Cheshkova, “A review of hyperspectral image analysis techniques for plant disease detection and identif ication,” Vavilov Journal of Genetics and Breeding, vol. 26, no. 2, p. 202, 2022.

25 L. Madden and G. Hughes, “Plant disease incidence: distributions, heterogeneity, and temporal analysis,” Annual Review of Phytopathology, vol. 33, no. 1, pp. 529– 564, 1995.

26 C. L. Campbell and L. V. Madden, Introduction to plant disease epidemiology.1990.

27 S. Wang, A. Khan, Y. Lin, Z. Jiang, H. Tang, S. Y. Alomar, M. Sanaullah, and U. A. Bhatti, “Deep reinforcement learning enables adaptive-image augmentation for automated optical inspection of plant rust,” Frontiers in Plant Science, vol. 14, p. 1142957, 2023.

28 M. N. Fawaiq, E. Utami, and D. Ariatmanto, “Rice plant disease detection with data augmentation using transfer learning,” Journal homepage: www. ijrpr. com ISSN, vol. 2582, p. 7421, 2023.

29 M. Veena and G. Bagewadi, “Identification of plant leaf disease using synthetic data augmentation program to improve the performance of deep learning models,” in International Conference on Engineering, Applied Sciences and System Modeling, pp. 185–199, Springer, 2017.

30 B. Sreenivasulu, A. Pasala, and G. Vasanth, “An effective data augmentation based on uncertainty based progressive conditional generative adversarial network for improving plant leaf disease classification.,” International Journal of Intelligent Engineering & Systems, vol. 16, no. 4, 2023.

31 G. Yilma, Z. Qin, M. Assefa, G. Alemu, and M. Ayalew, “Attention augmented convolutional neural network for fine-grained plant disease classification and visualization using stochastic sample transformations,” in Proceedings of the 5th International Conference on Advances in Image Processing, pp. 13–19, 2021.

32 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

33 Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

34 Tan, M. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

35 Teachable machine, 2024, from https://teachablemachine.withgoogle.com/