Open Access
Full Length Article

ORIGINAL ARTICLE

# Plant Disease Detection using Convolutional Neural Network

**Iram Noreen[1], Umar Farooq[2], Sehrish Ghaffar[3]**

[1] Department of Computer Science, Bahria University, Lahore Campus, Pakistan
[2] Department of Computer Science, Bahria University, Lahore Campus, Pakistan
[3] Department of Computer Science, Bahria University, Lahore Campus, Pakistan.

## ABSTRACT

Agriculture is the most important aspect of human life and also plays a vital role in a country's economy. Plants affected by diseases require huge effort, time, budget, and harmful sprays for treatment. The first symptoms of disease in plants are difficult to detect from the human eye at an early stage. In this study, fungus disease detection is performed using Convolutional Neural Network (CNN). This study focuses on detecting healthy or unhealthy plant leaves based on the very first fungus symptoms with high accuracy and an automated process. A pre-trained model VGG-16 using transfer learning is adopted, which is further fine-tuned and trained on the target plant's dataset for its upper layers. Further, a plant disease detection android application is provided for the user to detect fungus disease in plants. The android application uses the picture of the affected leaf to classify healthy and infected leaves. The proposed model showed 99% accuracy, which is better than other state-of-the-art methods.

**Keywords:** Plant Disease, Convolutional neural Network, Disease Detection, VGG-16

Author`s Contribution
*[1] Data analysis, interpretation, and manuscript writing, Active participation in data collection*
*[2] Conception, synthesis, planning of research*
*[3] Interpretation and discussion*

*Address of Correspondence*
*Iram Noreen*
*Email: Iram.bulc@bahria.edu.pk*

Article info.
*Received: July 6, 2021*
*Accepted: November 26, 2017*
*Published: December 30, 2021*

***Cite this article:*** *Noreen I, Farooq U, Ghaffar S.Plant Disease Detection using Convolutional Neural Network. J. inf. commun. technol. robot. appl.2021; 12(2):1-10.*

***Funding Source:*** *Nil*
***Conflict of Interest:*** *Nil*

## INTRODUCTION

A home garden represents multiple farming components such as trees, crops, shrubs, occasionally fisheries, herbs, and livestock, for adjoining the household on the land. A home garden not only provides fresh vegetables to the family but also provides business opportunities and increases the income of the household. Leaf diseases cause serious damage to plants if not addressed timely [1]. If gardening plants are unhealthy or weak then the nutrition process is affected. Many home garden owners know simple gardening tips; however, they have no way to early identify the plant's disease, hence could not adopt precautionary measures before a plant is visibly affected. An automated and easy method to inform home gardeners and farmers of the possible threat to plants and crops can save their pesticides budget, time, and effort to protect plants. In this contemporary era of technology, most research institutions are making scientific efforts for applying different techniques in home gardening for disease detection. The need of the hour is to employ scientific methods and computer automated applications to save the plants. Since the last decade, several image processing and machine learning-based approaches have been implied by the research community for disease detections in plants with several different scopes [2-4].

This study has the main focus of developing an intelligent mobile application based on machine learning and image processing technique to early identify plant diseases. The proposed method will be detecting a leaf with the name of a plant, the name of the disease, and its further guideline description. The proposed method is based on classifying a plant's leaf based on fungi disease and will result in a healthy or unhealthy detailed description. The smart application will not only identify the disease in plants but will also provide a remedy for the user to cure the disease. The proposed system will further inform the user about the growth rate and its care description. It will also provide a history of maintenance of plant-based on the initial discovery of disease patterns in the garden and recovery guidelines for further checking the schedule of a plant. The proposed system will be deployed as a smart application on an android phone for the end-user. This study will be beneficial for the users to get the disease updates and save their plants from the disease at an early stage of disease symptoms. The user application interfaces are simple and user-friendly. Fungus disease has many types, e.g., Downy Mildews, Powdery Mildews, Black Root, Clubroot, and Rusts [5]. This study has mainly focused on the black root and leaf-scorch disease in grapes and strawberries. This system detects the disease from the image captured by the mobile App and classifies the disease within a second with a high level of accuracy.

The rest of the paper is structured as follows. Section 2 describes Related Work in the domain. Section 3 describes materials and methods. Section 4 presents results and analysis. Section 5 presents the discussion of the results followed by a conclusion in Section 6.

## LITERATURE REVIEW

A lot of research work has been done in recent years for the early identification of fungi-based diseases in different sorts of fruit plants, vegetable plants, flowers, and crops [6]. Removal of green pixels, masking, thresholding, and evaluation of texture statistics are widely used techniques in the image processing domain for the detection of leaf diseases [7]. Radha et al. used an image processing technique based on thresholding and segmentation to identify the diseased region in the leaf [8]. However, this approach is highly dependent upon manual parameters. Gavhale et al. [9] introduced image processing techniques to detect leaf disease. They stored healthy and unhealthy leaves in the database. The images were captured with the help of an android phone camera and were sent to a classification server for diagnosis. Another challenge is that the introduction of noises furthers the image acquisition process of the leaf. The real image intensities are affected by the presence of noise that changes the pixel's value of an image. They further used the median filter for image enhancement and segmentation to segment the diseased region in the leaf image. After segmentation features extraction is used for extraction of the diseased portion. These features are helpful to identify the disease level. They further used an image enhancement process is to aid image analysis. Valliammai et al. [10] presented a method to reduce the image data by using the feature extraction process to enhance the features from the segmented image. However, image processing-based techniques are not fully automated for a vast range of plants.

Machine learning techniques have been widely used in this domain. Different researchers have applied machine learning for different plants, and with a different scopes of disease [11]. Most of the researchers have used the k-means and support vector machine algorithm on the dataset of citrus fruit, Rice, Maize, and banana for detecting fungi disease in plants [12, 13]. Singh and Mishra [6] used multiple machine learning methods Support Vector Machine (SVM), Perceptron Artificial Neural Network (ANN), and Genetic Algorithm (GA) for leaf disease detection. They extracted features from the input image and passed them to a classifier for classification. They resulted that SVM performed better in all approaches with 94.45% accuracy. Xi et al. [14] proposed an android-based system for diagnosing wheat disease. By using this system, users collect images of wheat diseases with the help of android phones and send images across the network to the server to diagnose the disease. Recently, Sharma et al. used KNN and SVM for early diagnosis of disease in rice plants and achieved very good performance with accuracies in the range of 77% to 98% for different scenarios [15]. Color and texture also play a very important role in machine learning methods. Shrivastava et al. extracted 172 features from a rice dataset with 14 color spaces using SVM with 94% accuracy [16].

In recent years deep learning methods have evolved with strong performance. Their performance in disease detection is also more sound than traditional machine learning methods due to their automated feature engineering and sparse links [17-22]. However, deep learning methods are highly dependent upon bulk data which has been a hinder in the plant's disease detection domain. Transfer learning is the key to this issue. Therefore, researchers are inclined towards using deep learning-based pre-trained models and they retrain these models according to different strategies on targeted datasets of plants using the transfer learning technique. Chowdhury et al. applied transfer learning using the EfficientNet deep learning model to detect disease in tomatoes leaves [23]. Nasir et al. used a fine-tuned VGG based transfer learning and acquired 99.06% accuracy for fruits diseases [24].

## RESEARCH METHODOLOGY

### Dataset

We have used the Plant Village dataset [25] for training and testing the proposed system. Plant Village dataset consists of 54303 leaf image samples providing healthy and unhealthy both types of categories. Data samples are divided into 38 different categories based on species and disease. Strawberry and grapes are two of the categories of this dataset. It provides more than 7000 images of grapes and strawberries for training and testing overall. The data samples are available in the format RGB.

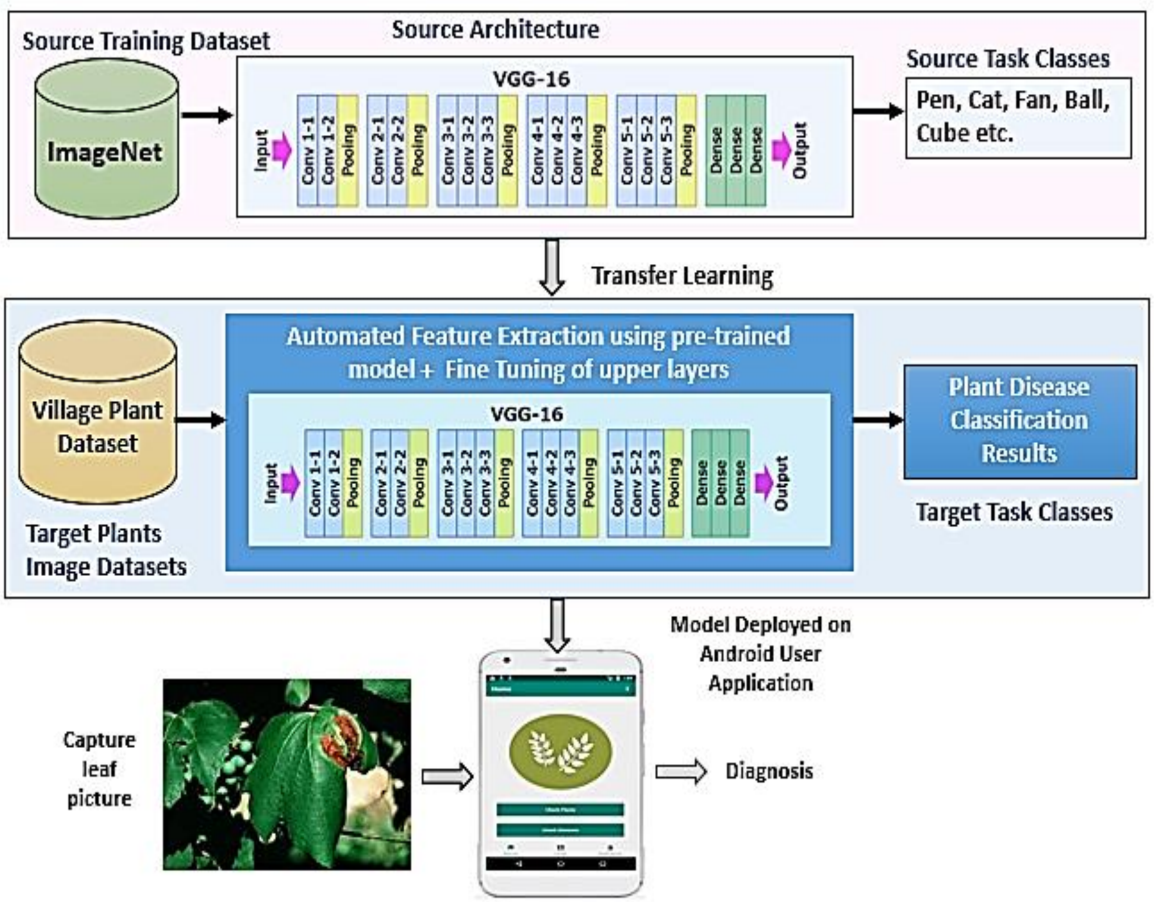



**Figure 1. Top Level Architecture**

### Architecture

The project methodology comprises two distinct phases. First, the phase performs training for plant leaf classification; second, the phase is related to disease detection and tracking via the smart application. The proposed system will be trained for two diseases in grapes and strawberry plants known as Black Rot and Leaf Scorch Convolutional Neural Network (CNN) based deep learning approaches [5] have reported good performance and accuracy for image and video data classification. Prominent features of CNN are automated feature extraction instead of manual process and invariance to affine transformations on images. However, deep learning approaches are hungry for computational cost, time complexity, and memory and data requirements. They unleash their true potential when they get huge size data set providing millions of data samples to train.

This study has resolved this issue by using the Transfer Learning technique also known as Knowledge Transfer. Transfer learning is a machine learning approach used for the improvement of a new task based on a pre-trained model architecture which is trained on a similar or another data model or problem domain. Several Pre-trained models exist which are based on CNN and are efficient in image data classification. We have used the VGG-16 model[26] in the proposed approach as a pre-trained model for transfer learning. It is an architecture of convolutional neural network which won the ImageNet Large Scale Visual Recognition Competition (ILSVR) in 2014. Therefore, it is considered one of the best CNN architectures for vision models. It has 16 weighted layers to train with approx. 138 million parameters. It has different categories of layers which include convolutional, max-pooling, activation, and Fully Connected (FC) layers. VGG16 is selected in this study because it does not require a large number of hyper-parameters for training. Instead, it focuses on convolution layers with a 3x3 filter with stride size 1 and pooling layers with a 2x2 filter, max function, and stride size 2. Moreover, it always uses image padding for its convolutional process. We have used VGG-16 trained on a popular and very large source dataset known as ImageNet [27]. Further, we trained it on our target Plant Village dataset [25]. We freeze all layers of VGG-16 except input and last classification layers for tuning, and output layers. We tuned the output layer according to the given classes. A top-level system of the proposed approach and the architecture of VGG-16 is shown in Figure 1 and Figure 2 respectively. The detail of its layers is described as follows.

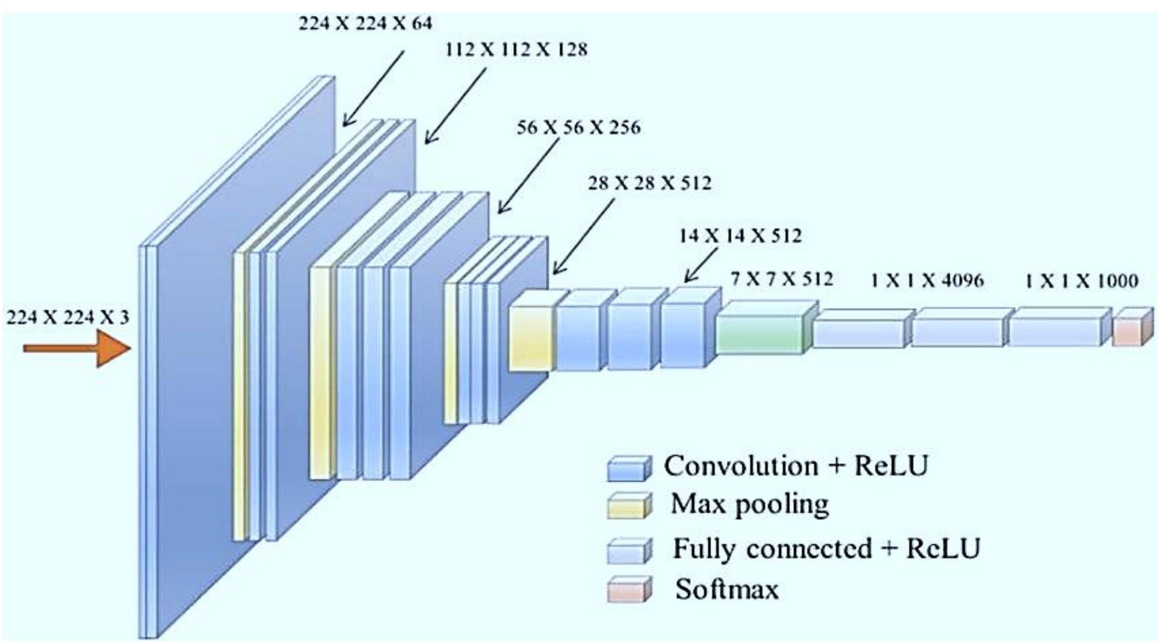



**Figure 3. The architecture of VGG-16 [28]**

***Input***: The input image size takes in RGB of 224x224 pixels, as shown in Figure 2.

*Convolutional layers:* In convolutional layers, 3x3 smallest size of the convolutional filter with a stride of 1 is used. Which captures left/right and up/down. 1x1 convolutional filters are also used which are followed by ReLU and act as a linear transformation of input.

***Max pooling layers***: Max-pooling used the window of 2×2 pixel with stride 2.

***Fully connected layers***: 3 fully connected layers are present in VGG. 4096 channels each present in the first 2 layers and 1000 channels 1 for each class in 3rd layer Image. We have kept frozen all hidden layers of VGG-16, however, the upper layers and output layer is trained on the plant's dataset images.

**Experiment setup**

We used Android studio (3.4.1) and Google Firebase to develop the front end of the smart application. Google COLAB cloud service is used for the development, training, and testing of the model because it is supported by Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs). Tensor flow libraries are used for the development and training of the model. We add our custom output layer of 4 classes shown in Table 1, with Softmax as the activation function shown in Figure 3. Detailed parameters and layers information is also shown in Figure 3. Following are the four classes in which the model is trained during experimentation.

**Table 1. Network Parameters**

| Parameters | Values |
|---|---|
| Classes | i. Grape Black_Rot<br>ii. Grape Healthy<br>iii. Strawberry Leaf Scorch<br>iv. Strawberry Healthy |
| Total Number of training images | 7178 |
| Total Number of testing images | 1795 |
| Total number of epochs | 5 |
| Steps per epoch | 225 |
| Total network parameters | 14,815,044 |
| Total Trainable parameters | 100,356 |
| Total Non-trainable parameters | 14,714,688 |

## RESULT AND ANALYSIS

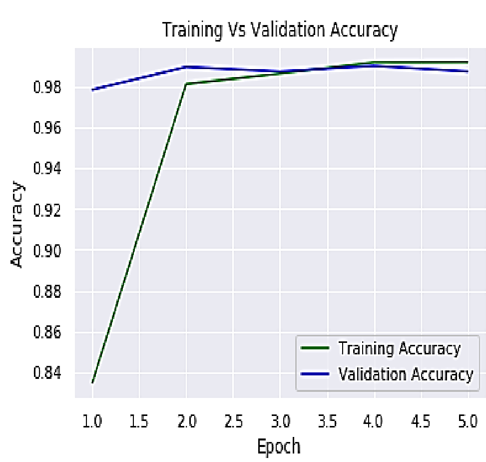
Training accuracy tells how well the data set is trained on the model. The accuracy score and plot for training and validation testing both are provided in Figure 4. When the number of epochs increases the training accuracy is also increased and after 4 epochs the training accuracy change at a constant rate. Validation accuracy tells how well a model is performed. When the number of epochs increases the validation accuracy is also increased and after 4 epoch the validation accuracy decrease. We get training accuracy of 99.25% and validation accuracy of 99.08%. Training loss is the error during training on the given data set. The number of epoch increase training loss starts decreasing. Validation loss is the error that occurs through the trained network after the validation of the data set. As the number of epoch increase validation loss starts decreasing. We get a Training Loss of 8.25% and a Validation Loss of 7.59%.

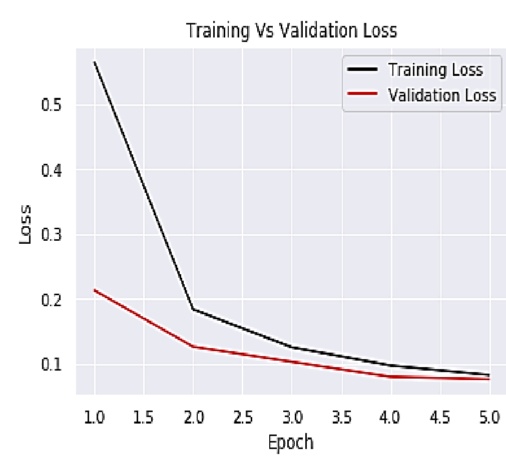
However, accuracy does not always reflect the true performance of the model. Therefore, other performance metrics such as Precision, Recall, and F1 Score are also used for the revaluation of performance. The performance metric used in the analysis of the results is presented in Eq. (1) – Eq. (4) and the score for each performance metric is shown in Figure 5. The confusion matrix shows the performance of the classifier on a given test data set for which the true values are known. The confusion matrix of 4 classes is shown in Figure 6.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_4 (InputLayer) | (None, 224, 224, 3) | 0 |
| block1_conv1 (Conv2D) | (None, 224, 224, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, 224, 224, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, 112, 112, 64) | 0 |
| block2_conv1 (Conv2D) | (None, 112, 112, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, 112, 112, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, 56, 56, 128) | 0 |
| block3_conv1 (Conv2D) | (None, 56, 56, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, 28, 28, 256) | 0 |
| block4_conv1 (Conv2D) | (None, 28, 28, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, 14, 14, 512) | 0 |
| block5_conv1 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv2 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv3 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_pool (MaxPooling2D) | (None, 7, 7, 512) | 0 |
| flatten_6 (Flatten) | (None, 25088) | 0 |
| dense_6 (Dense) | (None, 4) | 100356 |

Total params: 14,815,044
Trainable params: 100,356

**Figure 4. A model with customized output layers tuning parameters**



Recall = TP = TP / (TP+FN).......................................................................Eq. (1)
Precision = TP / (TP+FP).........................................................................Eq. (2)
Accuracy = (TP+TN) / (TP+TN+FP+FN)...............................................Eq. (3)
F1-Score = 2 * (Precision * Recall) / (Precision+ Recall)...........................Eq. (4)

```
Classification Report
                          precision    recall  f1-score   support

       Grape___Black_rot       0.99      1.00      0.99       472
         Grape___healthy       1.00      0.98      0.99       423
Strawberry___Leaf_scorch       1.00      0.97      0.99       444
    Strawberry___healthy       0.96      1.00      0.98       456

                accuracy                           0.99      1795
               macro avg       0.99      0.99      0.99      1795
            weighted avg       0.99      0.99      0.99      1795
```

**Figure 5. Classification report**

$$\begin{bmatrix} 470 & 2 & 0 & 0 \\ 4 & 414 & 0 & 5 \\ 0 & 0 & 432 & 12 \\ 0 & 0 & 0 & 456 \end{bmatrix}$$

**Figure 6. Confusion Matrix**

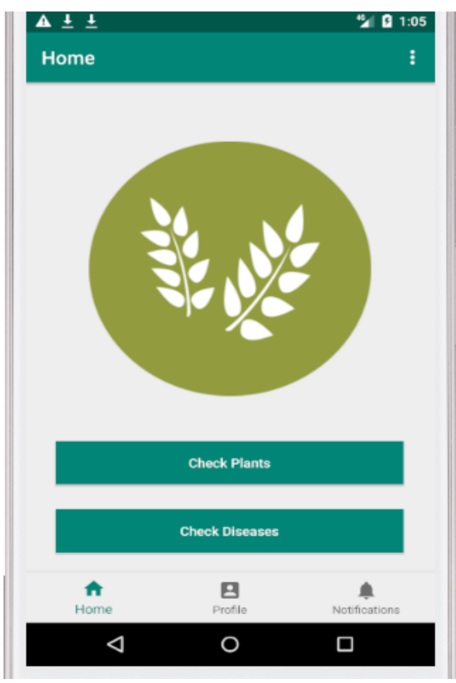
## DISCUSSION

A smart application interface is built using the Android platform. The interface of the smart application for the end-user is shown in Figure 7. The trained model is deployed on Android App, and it will be published on the platform for end-users. Moreover, a summary of comparison with state-of-the-art machine learning, and deep learning approaches is presented in Table 2. This study has used the VGG-16 method using fine-tuned transfer learning approach for black root and leaf-scorch disease detection in grapes and strawberries. The comparison shows that VGG-16 is well-tuned and trained to achieve better performance. However, this comparison is not specific and case-based. All other studies are conducted using different datasets for different plants and fruits. The performance is summarized just to showcase the general efficiency of this study and method.

**Table 2. A comparison with the State of the Art**

| No. | Author | Year | Approach | Accuracy |
|---|---|---|---|---|
| 1 | Prakash H. M. et al. [29] | 2015 | CNN | 90.96% |
| 2 | Iqbal Deep Kaur et al. [30] | 2016 | SVM | 96.77% |
| 3 | Sumit Nema et al [31] | 2017 | Bayes classifier and Fuzzy Logic | 94.45% |

| | | | | |
|---|---|---|---|---|
| 4 | Ehsan Kiania et al. [32] | 2017 | Fuzzy Logic Classification Algorithm (FLCA) | 97% |
| 5 | Faiza Anjum et al. [33] | 2018 | K-means segmentation + SVM | 96% |
| 6 | S.A. Khandekar et al. [34] | 2019 | SVM + K means clustering. | 92.4% |
| 8 | C. Deisy et al. [35] | 2019 | CNN | 87% |
| 10 | S.M. Jaisakthi et al. [36] | 2020 | Global Thresholding + SVM | 93% |
| 11 | Nasir et al. [24] | 2021 | VGG-19 with upper layers tuning and retraining | 99.06% |
| 12 | Proposed work | current | VGG-16 with upper layers tuning and retraining | 99.08% |



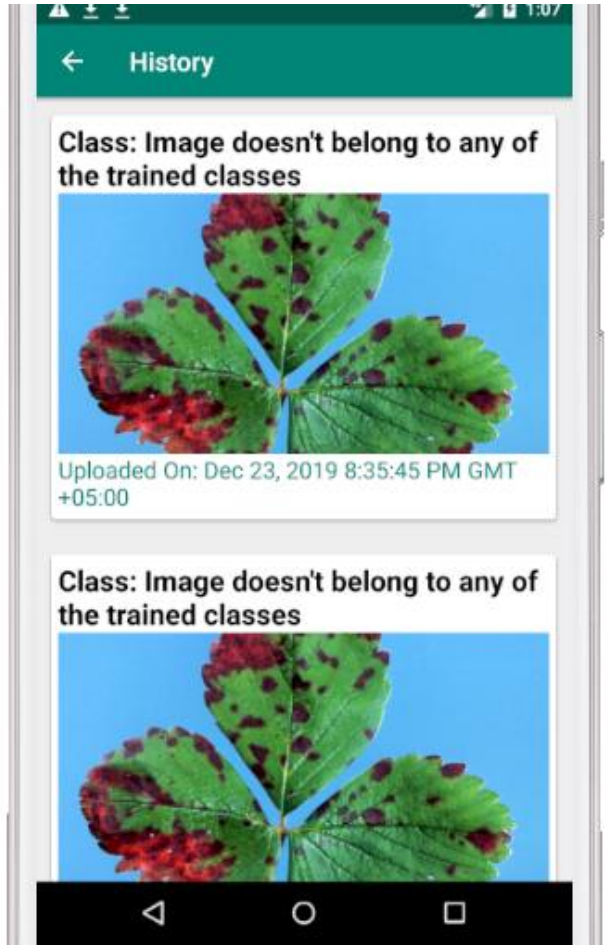




**Figure 7. The interface of the Android application for User**

## CONCLUSION

This study trained and tuned the VGG-16 network for the detection of plant diseases and also provided an Android application for the end-user. VGG-16 network is trained on the ImageNet dataset initially as the source dataset. Further, the model was trained on the plant's disease dataset and tuned to get optimum results on the grape and strawberry leaves. The diseases which are detected by using the application are Black Rot and Leaf Scorch. The results show that the proposed transfer learning model has achieved a training accuracy of 99.18% and validation accuracy of 98.72%. In the future, this application will be enhanced to detect more diseases like bacteria, and rust. Moreover, a comparative analysis with another pre-trained model can be performed in this problem domain.

## REFERENCES


1. López-Martínez, J., Juárez-Ramírez, R., Huertas, C., Jiménez, S., & Guerra-García, C. (2016, April). Problems in the adoption of Agile-Scrum methodologies: A systematic literature review. In Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in (pp. 141-148). IEEE.
2. P. A. Nazarov, D. N. Baleev, M. I. Ivanova, L. M. Sokolova, and M. V. Karakozova, "Infectious plant diseases: etiology, current status, problems and prospects in plant protection," Acta Naturae, vol. 12, p. 46, 2020.
3. U. Shruthi, V. Nagaveni, and B. Raghavendra, "A review on machine learning classification techniques for plant disease detection," in 2019 5th International conference on advanced computing & communication systems (ICACCS), 2019, pp. 281-284.
4. M. Loey, A. ElSawy, and M. Afify, "Deep learning in plant diseases detection for agricultural crops: a survey," International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), vol. 11, pp. 41-58, 2020.
5. S. S. Kumar and B. Raghavendra, "Diseases detection of various plant leaf using image processing techniques: A review," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 313-316.
6. H. A. Wael Alosaimi, M. Irfan Uddin, "PeachNet: Peach Diseases Detection for Automatic Harvesting," CMC-Computers, Materials & Continua, vol. 67, pp. 1665-1677, 2021.
7. D. V. Terentev A, Fedotov A, Eremenko D., "Current State of Hyperspectral Remote Sensing for Early Plant Disease Detection: A Review," Sensors, vol. 22, p. 757, 2022.
8. S. D. Khirade, & Patil, A. B., "Plant disease detection using image processing," in International conference on computing communication control and automation 2015, pp. 768-771.
9. S. Radha, "Leaf Disease Detection using Image Processing," J. Chem. Pharm. Sci., pp. 1-4, 2017.
10. K. R. G. a. U. Gawande, "An overview of the research on plant leaves disease detection using image processing techniques," IOSR J. Comput. Eng., vol. 16, pp. 10–16, 2014.
11. N. V. a. S. N. Geethalakshmi, "Multiple noise reduction using hybrid method for leaf recognition," in International Conference on Devices, Circuits and Systems (ICDCS), 2012, pp. 708–712.
12. L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, "Machine Learning for Plant Leaf Disease Detection and Classification–A Review," in 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0538-0542.
13. M. G. Selvaraj, A. Vergara, F. Montenegro, H. A. Ruiz, N. Safari, D. Raymaekers, et al., "Detection of banana plants and their major diseases through aerial images and machine learning methods: A case study in DR Congo and Republic of Benin," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 169, pp. 110-124, 2020.
14. K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize leaf disease detection and classification using machine learning algorithms," in Progress in Computing, Analytics and Networking, ed: Springer, 2020, pp. 659-669.
15. Y. L. Y. Xia, and C. Li, "Intelligent diagnose system of wheat diseases based on android phone," J. Inf. & Computational Sci., vol. 12, pp. 6845–6852, 2015.
16. C. J. K. A. D. Mayuri Sharma, "Early diagnosis of rice plant disease using machine learning techniques," Archives of Phytopathology and Plant Protection, vol. 55, pp. 259-283, 2022.
17. V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," Journal of Plant Pathology, vol. 103, pp. 17-26, 2021.
18. R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," Microprocessors and Microsystems, vol. 80, p. 103615, 2021.
19. H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," Data in brief, vol. 26, p. 104340, 2019.
20. G. Sambasivam and G. D. Opiyo, "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks," Egyptian Informatics Journal, vol. 22, pp. 27-34, 2021.
21. J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," Plant Methods, vol. 17, pp. 1-18, 2021.
22. G. Delnevo, R. Girau, C. Ceccarini, and C. Prandi, "A deep learning and social iot approach for plants disease prediction toward a sustainable agriculture," IEEE Internet of Things Journal, 2021.
23. M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, p. 468, 2019.
24. M. E. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, et al., "Automatic and reliable leaf disease detection using deep learning techniques," AgriEngineering, vol. 3, pp. 294-312, 2021.
25. I. M. Nasir, A. Bibi, J. H. Shah, M. A. Khan, M. Sharif, K. Iqbal, et al., "Deep learning-based classification of fruit diseases: An application for precision agriculture," 2021.
26. S. P. Mohanty, "Dataset of diseased plant leaf images and corresponding labels ", ed: GitHub-spMohanty/PlantVillage-Dataset.
27. O. Russakovsky, Deng, J., Su, H., et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, pp. 211–252, 2015.
28. L. Fei-Fei, Deng, J. and Li, K., "ImageNet: Constructing a large-scale image database," Journal of vision, vol. 9, pp. 1037-1037, 2009.
29. V. Pandiyan, T. Tjahjowidodo, W. Caesarendra, and P. Murugan, "In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning " Robotics and Computer-Integrated Manufacturing, vol. 57, pp. 477–487, 2019.
30. S. G. , P. M. M. , and M. A. , "Plant Leaf Disease Detection and Classification Using Image," International Journal of Innovative and Emerging, vol. 2, pp. 2394 – 5494, 2015.
31. I. K. , G. A. , and A. V. , "Detection and Classification of Disease Affected Region of Plant Leaves using Image Processing," Indian Journal of Science and Technology, vol. 9, December 2016.
32. S. N. , B. M. , and M. L. , "Advance App Design Methods of Leaf Disease Detection using Image Processing Approach – A Review," International Journal of Innovative Research in Science,Engineering and Technology, vol. 6, pp. 13277-13283, July 2017.
33. E. K. and T. M. , "Identification of plant disease infection using soft-computing: Application to modern botany," in 9th International Conference on Theory and Application of Soft

Computing, Computing with Words and Perception, Budapest, Hungary, 4-25 August 2017, pp. 893-900.
34. C. K. M. , F. A. , H. I. P. , and R. M. V. , "Plant Leaf Disease Identification System for Android," International Journal of Engineering Research in Computer Science and Engineering, vol. 5, pp. 48-53, June 2018.
35. A. a. K. Gargade, Mr SA, "A Review: Custard Apple Leaf Parameter Analysis and Leaf Disease Detection using Digital Image Processing," in 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 267-271.
36. M. a. D. Francis, C, "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks—A Visual Understanding," in 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 1063-1068.
37. S. a. M. Jaisakthi, P and Thenmozhi, D and others, "Grape Leaf Disease Identification using Machine Learning Techniques," in International Conference on Computational Intelligence in Data Science (ICCIDS), 2020, pp. 1-6.