Boosting Crop Yields: A Hybrid Approach to Intelligent Plant Disease Identification and Prediction

R. Shivali
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai, India

E. Elakiya
Assistant Professor
School of Computer Science and Engineering,
Vellore Institute of Technology, Chennai, India

B. Surendiran
Associate Professor
Department of Computer Science and Engineering, National Institute of Technology, Puducherry, Karaikal, India

ABSTRACT
Plant diseases are a major challenge for global food safety, and therefore it is impossible to underestimate the role of diagnostic methods. This paper promotes an integrated scheme that combines the capabilities of CNN and InceptionV3 models in order to diagnose plant disease. The proposed model integrates image processing algorithms, feature extraction techniques and ensemble learning in order to enhance accuracy and robustness. For evaluation purposes, we have used an all-inclusive dataset containing various ailments associated with corn maize rust, potato early blight, and tomato early blight. The dataset was divided into an 80-20 split ratio for training and testing respectively. Our findings are highly encouraging since the hybrid model recorded an accuracy level of 98.04%. Therefore, this research advances detection methodologies for plant ailments which could provide a dependable solution for use in agriculture. There is also future work that looks at tribrid models as well as comparison with existing literature to further enhance detection accuracy.

Keywords
Plant Disease, Deep Learning, Hybrid Model, Smart Farming

1. INTRODUCTION
Diseases in plants are a major threat to worldwide agriculture, which can affect agricultural yields, food security and economic stability. If left uncontrolled, these diseases caused by pathogens such as fungi, bacteria and viruses and other factors of the environment can cause huge crop losses. Plant–pathogen interrelationships are complicated and are influenced by fluctuating conditions that need advanced detection and management approaches. Rapid, accurate diagnosis is vital for identifying diseases in plants that can prompt measures to stop their spread & minimize financial impacts. To maintain a secure and stable global food supply it is necessary to deal with these problems. The growing concern over plant diseases endangering global food security has driven the development of sophisticated mechanisms of detection for timely intervention. This study investigates the use of DL models for plant diseases aiming at comparing its accuracy with other models. We aim at evaluating disease identification precision through integration with different techniques covering machine learning, image processing and biological knowledge. The crucial part of this project is the relative assessment among the existing models. In making such a comparison, we aim to identify what these models can do well or poorly. The outcomes of this study will not only help in coming up with methods for detecting diseases in plants but also result in us working on the development of a model that can accurately sense disease.

The model to be tested and trained is based on an extensive dataset that contain healthy and diseased leaves. This set includes diverse plant species and diseases which make it possible for the model to generalize results. Such a wide range of types enables the efficient identification of plants in different farming regions. It also helps improve its interpretational capacity by detecting image characteristics that are unique to specific illnesses. Thus, this design may eventually result in a more accurate disease diagnosis based on context with a detailed understanding of what is happening within crop tissues at the moment of capturing imagery from them by the model.

Moreover, this research seeks to build a bridge between the two by making practical suggestions for the reality. We anticipate that consulting agronomists and industrial partners will ensure our models are both precise and credible as well as easily adaptable to real field conditions. Additionally, this partnership will help bring in cutting edge technologies into current agricultural practices thereby benefiting farmers and other stakeholders in agriculture sector. Our aim is to make significant contributions towards plant disease detection and management, thus enhancing global food security through an interdisciplinary approach. To sum up, this research also aims at creating new technological prototypes besides using deep learning (DL) models for plant disease detection. In addition, the comparison with different models and exploring various approaches contribute to the advancement of precision, dependability and promptness of identifying a wide range of diseases on plants globally promoting the aim of ensuring food safety at large scale.

2. LITERATURE SURVEY
Traditional visual scouting methods [1] for detecting plant diseases often miss asymptomatic infections, necessitating the use of more advanced techniques like DNA-based or serological methods, such as PCR and ELISA, which are specific and sensitive but also laborious. Recent advancements have focused on combining soft computing with molecular techniques to improve detection efficiency. The integration of machine learning and nanotechnology presents a promising avenue for transforming infection detection systems and enhancing crop management practices while addressing food safety concerns.

Deep learning models have shown significant promise in classifying plant diseases from images, as demonstrated by a study using a 26-layer model on the BJFU100 dataset, achieving 91.78% accuracy [2]. However, challenges such as complex backgrounds and uneven lighting persist. Another study investigated various deep learning architectures using the PlantVillage dataset, finding that ResNet50 performed best under specific conditions, while networks like AlexNet and SqueezNet were less effective [3] [4]

The OMNCNN model, incorporating elements like ELM for classification and BF for preprocessing [20], achieved a high accuracy of 98.7%, outperforming other models like CNN-LVQ and VGG-16 [5]. Further research explored an optimized deep learning algorithm for herb plant disease classification, demonstrating superior performance metrics such as prediction rate and F-measure [6]. Data augmentation techniques and hyperparameter optimization, especially in ResNet-9, led to exceptional test accuracy and detection rates for specific diseases like blight in potatoes and tomatoes [7]

While both machine learning (ML) and deep learning (DL) techniques have been explored for plant disease detection, DL methods generally offer better performance, with models like VGG-16 achieving higher accuracy and precision compared to traditional ML approaches like Random Forest (RF) [8]. The development of mobile applications for disease diagnosis has also been suggested, allowing farmers to upload pictures for analysis. Future research aims to integrate IoT, cloud computing, and big data to enhance image processing capabilities and explore advanced classification methods [9].

Beyond image analysis, other advanced techniques include infrared thermal imaging for detecting temperature changes associated with diseases, as shown in studies on tea and tomato plants [10]. Spectroscopy and remote sensing technologies provide high spatial resolution for early infection detection, distinguishing between healthy and infected plants based on chlorophyll levels and other biomarkers [11]. Additionally, nucleic acid and protein analysis, along with mobility spectrometers and lateral flow devices, offer alternative methods for early disease detection [12]. Despite these advancements, there remains a need for accessible and efficient diagnostic tools that can be used widely in agricultural practices [13].

3. MATERIALS AND METHOD

3.1 Proposed Model

The proposed methodology for plant disease detection employs a hybrid model approach that integrates deep learning techniques. The process begins with dataset acquisition and pre-processing, followed by partitioning the data into training, validation, and testing sets. The dataset includes approximately 87,000 images of healthy and diseased crop leaves, sourced from the new plant disease dataset and augmented from the original PlantVillage dataset. For this study, only three disease categories are considered: Corn Maize - Common Rust, Potato - Early Blight, and Tomato - Early Blight[15]. The dataset is split into 80% for training and 20% for validation, while a separate directory is maintained for testing images. This structure supports effective training and testing of the detection models. During model selection, two deep learning models are chosen, and their feature extraction layers are frozen. The features extracted from both models are flattened and concatenated into a single feature vector, forming a hybrid model that learns from both sets of features. A dense layer with 512 units and ReLU activation, followed by a dropout layer, is employed to process these features and prevent overfitting. The final dense layer, with softmax activation, outputs the probabilities for each class, with training focused on minimizing the categorical cross-entropy loss.

In the evaluation phase, the hybrid model's performance is assessed using several metrics, including training accuracy, validation accuracy, training loss, and validation loss. Training accuracy reflects the percentage of correct predictions on the training samples, while validation accuracy indicates the model's accuracy on the validation set. Similarly, training loss measures the average error over all training samples, and validation loss gauges how well the model generalizes to new, unseen data. These metrics collectively provide insight into the model's ability to accurately detect and classify plant diseases, highlighting its effectiveness and potential areas for improvement represented in Figure 1.

Table 1. Plant disease dataset

Disease	Train	Validation
Corn Maize – Healthy	1859	465
Corn Maize – Common Rust	1907	477
Potato – Healthy	1824	456
Potato – Early Blight	1939	485
Tomato – Healthy	1926	481
Tomato – Early Blight	1920	480
Total	**11375**	**2846**

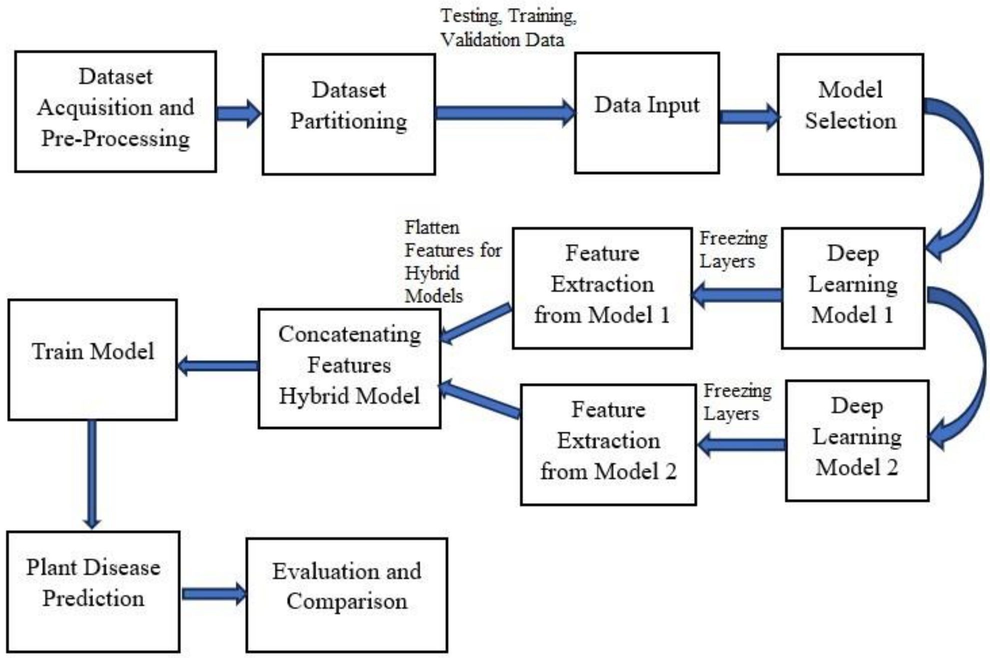

Fig 1: Proposed Model

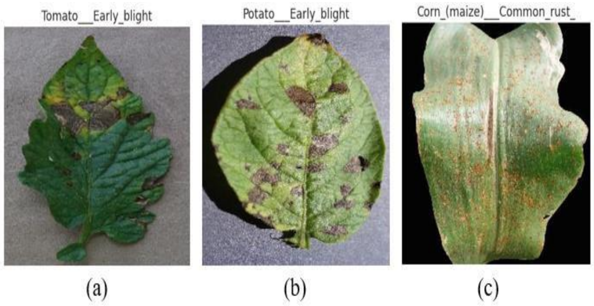

Fig 2: (a) Tomato Early Blight (b) Potato Early Blight (c) Corn Maize Common Rust

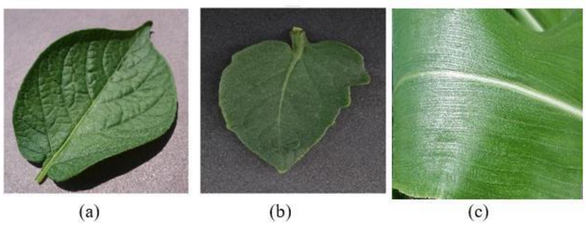

Fig 3: (a) Tomato Healthy (b) Potato Healthy (c) Corn Maize Healthy

3.2 Theoretical Background

3.2.1 CNN

CNNs are the most commonly used deep learning algorithms in image classification and analysis. Three main types of layers constitute CNNs: pooling, fully connected, and convolutional layers. Convolutional and Pooling Layers play a vital part in feature extraction from input data [16]. This is the layer that extracts different features from images. When going deeper into the network, it extract more complex information such as object orientation or texture whereas at its initial stages, learns trivial properties like edges and boundaries [19].

3.2.2 InceptionV3

InceptionV3, a convolutional neural network (CNN) model, is designed with several key layers for efficient image data processing, starting with an input layer that accepts three-channel RGB images. The model's core consists of convolutional layers that extract features from the input data, utilizing various filter sizes within inception modules to capture both fine-grained and coarse-grained information simultaneously. Pooling layers reduce the spatial dimensions of feature maps, retaining essential information, while batch normalization layers stabilize and accelerate training. Activation functions like ReLU introduce non-linearity, enabling the network to learn complex features [22].

3.2.3 MobileNet

MobileNet is a convolutional neural network (CNN) architecture designed for devices with limited computational power, offering efficiency and speed ideal for low-power, real-time processing. Its standout feature is the use of depth-wise separable convolutions, which split regular convolutions into depth-wise and pointwise convolutions. This separation significantly reduces computational costs while maintaining high accuracy. MobileNet's lightweight architecture, with fewer parameters than traditional CNNs, enhances its speed and efficiency. It also includes components like batch normalization to accelerate learning, rectified linear unit activation functions for non-linearity, and down sampling layers that reduce feature map sizes, retaining essential information.

3.2.4 ResNet50

ResNet50, a deep neural network architecture, is constructed by stacking multiple layers and commonly uses a 3-channel RGB image format in its input layer. Its key feature is the use of residual blocks, which include skip connections to mitigate the vanishing gradient problem in deep networks, enabling the training of deeper architectures. These blocks vary in the number of layers to capture different levels of abstraction.

ResNet50 employs pooling layers to reduce spatial dimensions and retain meaningful information, while batch normalization layers stabilize and accelerate the training process, despite occasional instability. The network introduces non-linearity through activation functions like ReLU, allowing it to recognize complex patterns in the input data [17].

3.2.5 VGG16

VGG16 is a straightforward yet effective convolutional neural network (CNN) architecture [14] consisting of 16 layers, including convolutional layers, pooling layers, fully connected layers, and an output layer. It processes 224x224 RGB images as input. The convolutional layers use small 3x3 filters with a stride of 1 pixel to extract detailed features from the images. Max pooling layers then reduce the spatial dimensions of the feature maps by selecting the maximum value from each group of pixels, retaining the most relevant information [23].

3.3 Hybrid Models

3.3.1 Inception - CNN

The hybrid model architecture combines the InceptionV3 architecture for feature extraction with a customized CNN model, leveraging the pre-trained weights from ImageNet for the InceptionV3's top classification layers. With an input shape of (224, 224, 3), this hybrid model utilizes the multi-scale information extracted by InceptionV3 and passes it to subsequent CNN layers for further processing and classification. Initially, only the last four layers of each model were unfrozen to allow fine-tuning during training, while the rest retained their pre-trained weights, preserving the learned characteristics. Both models were then trained to produce flat outputs for concatenation, ensuring effective integration of features. This integration enables the hybrid model to learn diverse feature representations and reduces sensitivity to noise. The concatenated features are processed by two dense layers with ReLU activation functions and dropout layers to prevent overfitting. The final layer uses a softmax activation function to calculate class probabilities, categorizing input images into one of six specified classes.

Algorithm

InceptionV3 and CNN for Plant Disease Detection

Input: X
d: dataset of RGB images of plant leaves
l: true labels for the images
Output: Score obtained for the hybrid model on the test dataset

For each epoch do:
  #CNN Feature Extraction
    For each convolution layer do:
      For each sample in X do:
        Calculate a_{ij}^{m} from X by the convolution layer process
      End for
#Dimensions of a is (512 – KernelSize + 1, FilterSize)
        If a_{ij}^{m} length < 512 do:
          Apply zero padding to a_{ij}^{m}
          #Dimension of a is (512, FilterSize)
      End If
    End For
    #Dimension of a is (512, 512)
#InceptionV3 Feature Extraction
    For each convolution layer do:
      For each sample in X do:
        Apply Inception Module on X
        Calculate a_{ij}^{m} from X
      End for
    #Dimension of a is (1, num_filters)
    End for

#Hybrid Model
let fet be the feature set of images in d
for each image img in data do
    preprocess img for input into CNN and Inception
end for
train_fet, test_fet, train_label, test_label split feature set and labels into train and test subset
train and test the Inception model
M_inception <- InceptionModel (train_fet, train_label)
inception_train <- M_inception.predict(train_fet)
inception_test <- M_inception.predict(test_fet)
train and test the CNN model
M_CNN <- CNNModel(train_fet, train_label)
CNN_train <- M_CNN.predict(train_fet)
CNN_test <- M_CNN.predict(test_fet)
model_train <- concatenate (CNN_train, inception_train)
model_test <- concatenate (CNN_test, inception_test)
merged_model <- Dense (inception_train, CNN_train)
score <- evaluate(model_test, test_label)
return score

3.3.2 Inception - MobileNet

The hybrid model architecture merges features from MobileNet and InceptionV3 for image classification, leveraging the strengths of both CNN architectures to enhance overall performance. This approach involves unfreezing the last four layers of both MobileNet and InceptionV3 models for fine-tuning, while keeping the remaining layers frozen with their pre-trained weights [21]. This strategy allows the model to adapt to specific dataset characteristics while retaining the beneficial features learned from the original models. Once unfreezing is complete, the output features are flattened and merged into a unified feature vector, which is then processed through two dense layers with ReLU activation functions and dropout layers to mitigate overfitting. The final layer uses a softmax activation function to estimate class probabilities, categorizing input images into predefined classes. This combined architecture offers improved accuracy and robustness in classification by harnessing the diverse strengths of both MobileNet and InceptionV3.

Algorithm

InceptionV3 and MobileNet for Plant Disease Detection
Input: X
d: dataset of RGB images of plant leaves
l: true labels for the images
Output: Score obtained for the hybrid model on the test dataset

For each epoch do:
  #MobileNet Feature Extraction
    For each convolution layer do:
      For each sample in X do:
        apply depth wise convolution layer
        apply point wise convolution layer
      End for
    # Dimension of a is (1, num_filters)
      If a_{ij}^{m} length < 512 do:
        Apply zero padding to a_{ij}^{m}
Dimension of a is (512, FilterSize)

End If
End For
#Dimension of a is (512, 512)

#InceptionV3 Feature Extraction
For each convolution layer do:
For each sample in X do:
Apply Inception Module on X
Calculate a_{ij}^{m} from X
End for
#Dimension of a is (1, num_filters)
End for

#Hybrid Model
let fet be the feature set of images in d
for each image img in data do
preprocess img for input into Inception and MobileNet
end for
train_fet, test_fet, train_label, test_label split feature set and labels into train and test subset
train and test the Inception model
M_inception <- InceptionModel (train_fet, train_label)
inception_train <- M_inception.predict(train_fet)
inception_test <- M_inception.predict(test_fet)
train and test the MobileNet model
M_mobile <- MobileNetModel(train_fet, train_label)
mobile_train <- M_mobile.predict(train_fet)
mobile_test <- M_mobile.predict(test_fet)
model_train <- concatenate (mobile_train, inception_train)
model_test <- concatenate (mobile_test, inception_test)
merged_model <- Dense (inception_train, mobile_train)
score <- evaluate(model_test, test_label)
return score

3.3.3 Inception - ResNet50

The hybrid model integrates features from the powerful Inception and ResNet50 architectures, utilizing residual connections for enhanced training. Initially, the last four layers of each model are unfrozen for fine-tuning, while earlier layers remain fixed with their pre-trained weights. After unfreezing, outputs from both models are flattened and concatenated, enabling the model to combine extracted features effectively. This fusion captures a broader range of information from input images, enhancing the model's ability to learn discriminative representations. The concatenated features are then processed through additional layers, including two dense layers with ReLU activation functions, which refine the features for classification. Dropout layers are included after each dense layer to prevent overfitting. The final output layer uses a softmax activation function to predict class probabilities, categorizing images into one of six predefined classes. By combining Inception and ResNet50, this hybrid approach aims to improve generalization and predictive accuracy on unseen data

Algorithm

.InceptionV3 and ResNet50 for Plant Disease Detection
Input: X
d: dataset of RGB images of plant leaves
l: true labels for the images
Output: Score obtained for the hybrid model on the test dataset

For each epoch do:
#ResNet50 Feature Extraction
For each convolution layer do:
For each sample in X do:
Apply convolution layer 1, 2, 3
Apply Shortcut Connection
End for
#Dimensions of a is (height, width, FilterSize)
If a_{ij}^{m} length < 512 do:
Apply zero padding to a_{ij}^{m}
#Dimension of a is (512, FilterSize)
End If
End For
#Dimension of a is (512, 512)

#InceptionV3 Feature Extraction
For each convolution layer do:
For each sample in X do:
Apply Inception Module on X
Calculate $aijm$ from X
End for
#Dimension of a is (1, num_filters)
End for

#Hybrid Model
let fet be the feature set of images in d
for each image img in data do
preprocess img for input into Inception and MobileNet
end for
train_fet, test_fet, train_label, test_label split feature set and labels into train and test subset
train and test the Inception model
M_inception <- InceptionModel (train_fet, train_label)
inception_train <- M_inception.predict(train_fet)
inception_test <- M_inception.predict(test_fet)
train and test the ResNet model
M_resnet <- ResNetModel(train_fet, train_label)
resnet_train <- M_resnet.predict(train_fet)
resnet_test <- M_resnet.predict(test_fet)
model_train <- concatenate (resnet_train, inception_train)
model_test <- concatenate (mobile_test, inception_test)
merged_model <- Dense (resnet_train, inception_train)
score <- evaluate(model_test, test_label)
return score

3.3.4 Inception - VGG16

The proposed hybrid model integrates features from the InceptionV3 and VGG16 architectures, aiming to leverage their unique strengths for enhanced image classification performance. This approach involves unfreezing the last four layers of both models for fine-tuning, while keeping the remaining layers frozen with their pre-trained weights. This allows the model to adapt to the specific dataset while still benefiting from the robust features extracted by the original models. After unfreezing, the outputs from both models are flattened and concatenated into a single combined feature vector. This vector is then passed through a dropout layer to reduce overfitting, followed by two dense layers with ReLU activation functions to refine the features for better classification. The final layer uses a softmax activation function to predict class probabilities, categorizing images into one of six predefined classes. By combining the distinctive characteristics of InceptionV3 and VGG16, this hybrid model aims to achieve improved performance and generalization, potentially leading to more accurate and robust classification outcomes.

Algorithm

InceptionV3 and VGG16 for Plant Disease Detection
Input: X
d: dataset of RGB images of plant leaves
l: true labels for the images
Output: Score obtained for the hybrid model on the test dataset

1: For each epoch do:
2: #VGG16 Feature Extraction
3: For each convolution layer do:
4: For each sample in X do:
5: Apply convolution
End for
6: #Dimensions of a is (height, width, FilterSize)
7: If a_{ij}^{m} length < 512 do:
8: Apply zero padding to a_{ij}^{m}
9: #Dimension of a is (512, FilterSize)
10: End If
11: End For
12: #Dimension of a is (512, 512)

13: #InceptionV3 Feature Extraction
14: For each convolution layer do:
15: For each sample in X do:
16: Apply Inception Module on X
17: Calculate a_{ij}^{m} from X
18: End for
19: #Dimension of a is (1, num_filters)
20: End for

#Hybrid Model
21: let fet be the feature set of images in d
22: for each image img in data do
23: preprocess img for input into Inception and MobileNet
24: end for
25: train_fet, test_fet, train_label, test_label split feature set and labels into train and test subset
26: train and test the Inception model
27:M_inception <- InceptionModel (train_fet, train_label)
28: inception_train <- M_inception.predict(train_fet)
29: inception_test <- M_inception.predict(test_fet)
30: train and test the VGG model
31: M_VGG <- VGGModel(train_fet, train_label)
32: VGG_train <- M_VGG.predict(train_fet)
33: VGG_test <- M_VGG.predict(test_fet)
34:model_train <- concatenate (VGG_train, inception_train)
35: model_test <- concatenate (VGG_test, inception_test)
36: merged_model <- Dense (inception_train, VGG_train)
37: score <- evaluate(model_test, test_label)
38: return score

4. RESULTS AND DISCUSSION

Table 2 displays the accuracy scores achieved by various hybrid models in the plant disease detection task. The accuracies of hybrid models are comparative to those of single models when they are made by combining features from multiple pre-trained models. This would imply that overall performance can be improved if we leverage on the capabilities of different architectures through model fusion. The fact that these hybrid models have higher accuracies than others which involve InceptionV3 and MobileNet implies that these two networks are good at capturing diverse and complementary features, thus improving their detection capacity. Nonetheless, compared with separate implementations, InceptionV3-ResNet50 is a better choice than Resnet50 with low accuracy. This may lead to slight decrease of accuracy by fused model such as InceptionV3-ResNet50 or less accurate individual model ResNet50. Pretrained architecture provides a new opportunity for detecting diseases due to their collective ability.

Table 2. Accuracy Comparison of used models

Model	Accuracy
InceptionV3+CNN	95.32
InceptionV3+Resnet50	94.96
InceptionV3+MobileNet	98.40
InceptionV3+VGG16	97.84

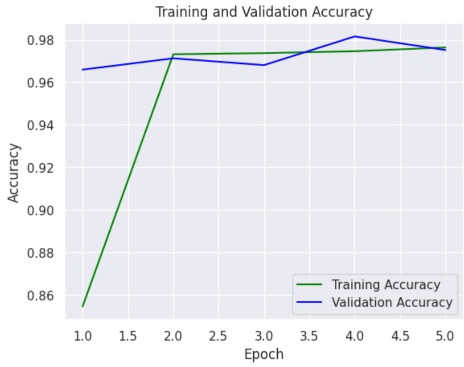

Fig. 4. InceptionV3 Accuracy Graph

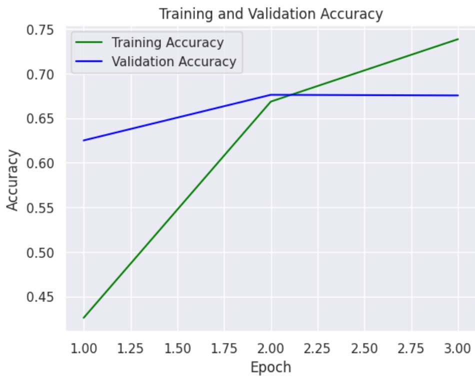

Fig. 5. ResNet50 Accuracy Graph

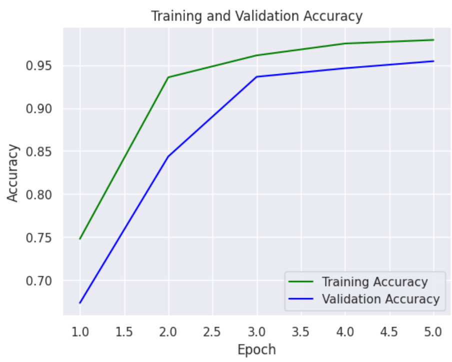

Fig. 6. CNN Accuracy Graph

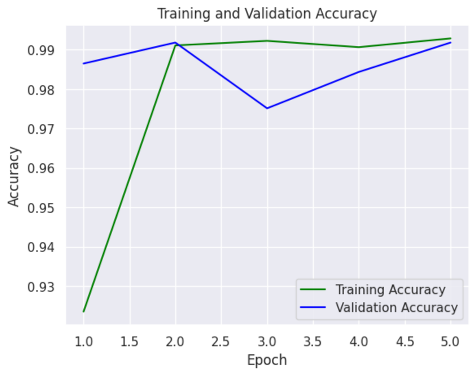

Fig. 7. MobileNet Accuracy Graph

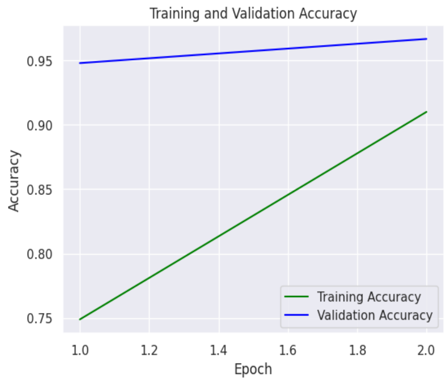

Fig. 8. CNN + InceptionV3 Accuracy Graph

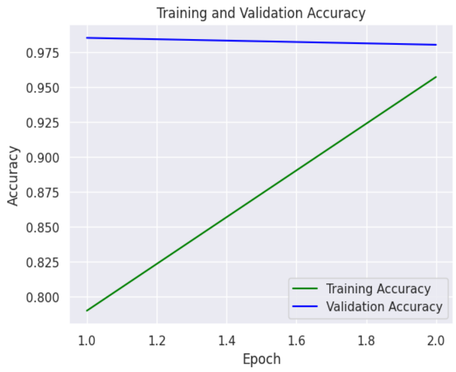

Fig. 9. VGG16 + InceptionV3 Accuracy Graph

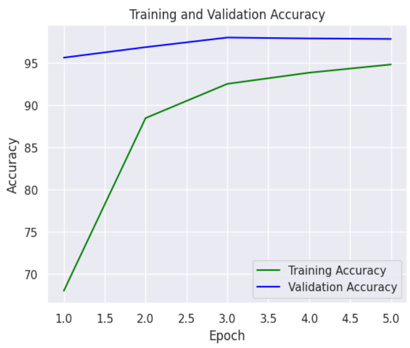

Fig.10. ResNet50 + InceptionV3 Accuracy Graph

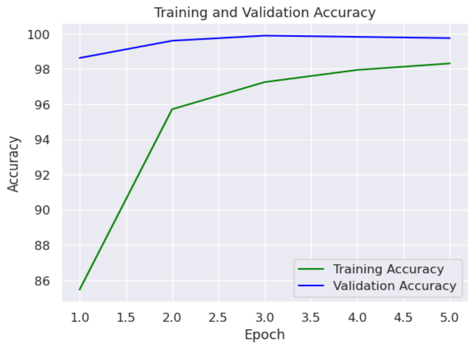

Fig. 11. MobileNet + InceptionV3 Accuracy Graph

5. CONCLUSION AND FUTURE WORK

The study illustrates the effectiveness of various pretrained models and hybrid methodologies in the detection of plant diseases. Our findings indicate that models such as MobileNet achieve notable accuracies, suggesting their suitability for this application. Additionally, hybrid models, particularly those combining InceptionV3, demonstrate competitive performance, underscoring the potential of model fusion techniques.

In future research, we aim to explore several avenues for further improvement. This includes enhancing the hybrid models by integrating diverse architectures and exploring alternative fusion strategies. Furthermore, expanding the dataset to encompass a wider range of plant species and disease classes could enhance the generalization capabilities of the models. Additionally, there is potential to develop a user-friendly web application for real-time plant disease detection, leveraging the trained models for practical agricultural applications [18]. This study contributes to the advancement of methodologies for detecting plant diseases and provides a basis for future research in this field. Future research could involve comparing our results with alternative approaches that address the challenges faced in this study, as well as exploring enhancements to improve the efficiency of plant disease detection.

6. REFERENCES

[1] Attaluri, S., Dharavath, R. Novel plant disease detection techniques-a brief review. Mol Biol Rep 50, 9677–9690 (2023)

[2] Yu Sun, Yuan Liu, Guan Wang, Haiyan Zhang, "Deep Learning for Plant Identification in Natural Environment", Computational Intelligence and Neuroscience, vol. 2017, Article ID 7361042, 6 pages, 2017

[3] C Jackulin, S. Murugavalli, A comprehensive review on detection of plant disease using machine learning and deep learning approaches, Measurement: Sensors, Volume 24, 2022, 100441, ISSN 2665-9174

[4] Sachin Dahiya, Tarun Gulati, Dushyant Gupta, Performance analysis of deep learning architectures for plant leaves disease detection, Measurement: Sensors, Volume 24, 2022, 100581, ISSN 2665-9174

[5] S. Ashwinkumar, S. Rajagopal, V. Manimaran, B. Jegajothi, Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks, Materials Today: Proceedings, Volume 51, Part 1, 2022, Pages 480-487, ISSN 2214-7853

[6] V. Sathiya, M.S. Josephine, V. Jeyabalaraja, An automatic classification and early disease detection technique for

herbs plant, Computers and Electrical Engineering, Volume 100, 2022, 108026, ISSN 0045-7906

[7] Alberta Odamea Anim-Ayeko, Calogero Schillaci, Aldo Lipani, Automatic blight disease detection in potato (Solanum tuberosum L.) and tomato (Solanum lycopersicum, L. 1753) plants using deep learning, Smart Agricultural Technology, Volume 4, 2023, 100178, ISSN 2772-3755

[8] R. Sujatha, Jyotir Moy Chatterjee, NZ Jhanjhi, Sarfraz Nawaz Brohi, Performance of deep learning vs machine learning in plant leaf disease detection, Microprocessors and Microsystems, Volume 80, 2021, 103615, ISSN 0141-9331

[9] Mohanty Sharada P. , Hughes David P. , Salathé Marcel, Using Deep Learning for Image-Based Plant Disease Detection, Frontiers in Plant Science, Volume 7, 2016

[10] Zou, X.; Ren, Q.; Cao, H.; Qian, Y.; Zhang, S. Identification of Tea Diseases Based on Spectral Reflectance and Machine Learning. J. Inf. Process. Syst. 2020, 16, 435–446

[11] El-Sayed Ewis Omran (2017) Early sensing of peanut leaf spot using spectroscopy and thermal imaging, Archives of Agronomy and Soil Science, 63:7, 883-896

[12] Martinelli, F., Scalenghe, R., Davino, S. et al. Advanced methods of plant disease detection. A review. Agron. Sustain. Dev. 35, 1–25 (2015)

[13] Fang, Y.; Ramasamy, R.P. Current and Prospective Methods for Plant Disease Detection. Biosensors 2015, 5, 537-561

[14] Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z., & Jasińska, E. (2021, June 9). Identification of plant-leaf diseases using CNN and Transfer-Learning Approach. MDPI

[15] Konstantinos P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture, Volume 145, 2018, Pages 311-318.

[16] Trivedi, N. K., Gautam, V., Anand, A., Aljahdali, H. M., Villar, S. G., Anand, D., Goyal, N., & Kadry, S. (2021, November 30). Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network

[17] Qi, H., Liang, Y., Ding, Q., & Zou, J. (2021, February 23). Automatic identification of peanut-leaf diseases based on Stack Ensemble. MDPI.

[18] G. Geetha et al 2020, Plant Leaf Disease Classification and Detection System Using Machine Learning

[19] G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020, pp. 109-113

[20] Elakiya E., Rajkumar N. Designing preprocessing framework (ERT) for text mining application 2017 Int. Conf. IoT Appl, IEEE (2017), pp. 1-8, 10.1109/ICIOTA.2017.8073613

[21] P. Sharma, R. K. Yadav and S. S. Rawat, "Hybrid Models for Plant Disease Detection using Transfer Learning Technique," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 712-718

[22] Mishalee Lambat, Rushab Kothari, Mitrup Kabi, Tushar Mane, “Plant Disease Detection Using InceptionV3”, Jun 2022, International Research Journal of Engineering and Technology (IRJET), Volume: 09 Issue: 06.

[23] Alok Kumar, Ankit Kumar, “Plant Disease Detection using VGG16”, 2023 IJCRT, Volume 11, Issue 1 January 2023 | ISSN: 2320-2882