IJETIR

ISSN (Online) 2583-0554

International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 5, Issue 5, May 2025

Impact Factor: 5.731

Machine Learning Based Leaf Disease Detection System

Ms. Sangeetha P[1] and Dr Kavitha P[2]

II MCA Final Year, Department of Computer Applications[1]

Assistant Professor, Department of Computer Applications[2]

School of Computing Sciences, Vels Institute of Science Technology and Advanced Studies (VISTAS), Chennai

sangeetha080703@gmail.com and pkavikamal@gmail.com

Abstract: *Agriculture is vital for global food production, and keeping crops healthy is very important. One big challenge farmers face is detecting plant leaf diseases early, before they spread and cause major damage. Checking plants manually is slow, tiring, and often not accurate. To solve this, we developed an automated system that uses artificial intelligence (AI), especially machine learning and deep learning, to quickly find diseases in plant leaves by analyzing images.*

Our system uses a method called YOLO (You Only Look Once), which is very fast at spotting objects in pictures. When an image is given to the system, it passes through several steps — first the image is read, then important features are picked out, possible disease areas are marked with boxes, and finally, the best results are shown after removing any overlapping guesses. We trained the model using many labeled pictures of leaves, and made it smarter using optimization methods. The final model achieved 81% accuracy when tested, meaning it can correctly detect diseases most of the time.

We built the system using popular tools like TensorFlow and Python, and we worked on it using Google Colab, a cloud platform. The system can also organize files automatically, saving processed images and creating folders when needed, making it easy to use on a bigger scale.

This project shows how AI can help farmers by providing a quick, affordable, and reliable tool for identifying plant diseases early. It can help improve crop health, reduce losses, and support better farming practices for the future.

Keywords: Plant Disease Detection, Agriculture, Artificial Intelligence(AI), Machine Learning, Deep Learning, Image Analysis, YOLO Algorithm, Leaf Health Monitoring, Automated Detection System, Precision Farming

I. INTRODUCTION

Farming is the backbone of food production, and keeping crops healthy is essential for ensuring good yields. However, plant diseases, especially those that first appear on leaves, are one of the major problems that farmers face. If these diseases are not caught early, they can spread across entire fields, leading to major financial losses and even food shortages. Traditionally, farmers or agricultural experts inspect plants manually, walking through large fields to check for visible signs of infection. This method is slow, tiring, and sometimes inaccurate, especially when early symptoms are very small or hidden.

In today's world, technology offers better ways to handle these challenges. Artificial intelligence, machine learning, and image recognition techniques are making it possible to monitor plant health more effectively and in less time. This project introduces a **Leaf Disease Detection System** that uses deep learning to automatically identify diseases from images of plant leaves. Instead of relying on manual observation, the system processes leaf images and instantly highlights any signs of disease.

At the core of the system is the **YOLO (You Only Look Once)** object detection algorithm, known for its speed and accuracy. YOLO can quickly scan an image, find areas that look diseased, and provide results in real-time. The model

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

has been trained using a large collection of labeled leaf images to recognize different disease patterns. It uses techniques like bounding box prediction and non-maximum suppression to deliver clean and accurate results.

By applying this system, farmers can save time, reduce mistakes, and respond quickly to disease outbreaks. It not only helps protect crops but also supports more sustainable farming by reducing unnecessary pesticide use. Overall, the Leaf Disease Detection System shows how combining agriculture with AI can create smarter, faster, and more reliable farming practices for the future.

II. EXISTING RESEARCH

In the past few years, scientists and engineers have been working to find better ways to detect plant diseases early. Traditionally, farmers would check plants manually for signs of disease, but this method is often slow, tiring, and not always accurate — especially when the disease is just beginning and hard to spot.

At first, researchers used basic computer programs that looked at colors, shapes, and textures of leaves to guess if a plant was healthy or sick. They combined this with simple machine learning models like **SVM**, **KNN**, and **Decision Trees**. These methods did work to some extent, but they needed a lot of manual work to figure out which parts of the image were important. Plus, they often struggled when the images were taken in different lighting or had different backgrounds, like what happens on real farms.

Things changed when **deep learning** came into the picture, especially with the use of **Convolutional Neural Networks (CNNs)**. These networks could learn by themselves from the images without needing anyone to manually choose features. For example, one study by **Mohanty and his team** in 2016 showed that CNNs could detect plant diseases with over **99% accuracy** using a large collection of leaf images called **PlantVillage**. However, these systems could only **say** what disease was present — they couldn't **show** exactly where the disease appeared on the leaf.

To improve this, researchers started using **object detection models** like **YOLO**, **Faster R-CNN**, and **SSD**. These models could both find the disease on the leaf and also tell what kind of disease it was. Among them, **YOLO** became very popular because it is **fast, accurate**, and can be used on smartphones or small devices. Newer versions like **YOLOv5** and **YOLOv8** made the system even quicker and more reliable.

Researchers also added techniques like **data augmentation** (to create more training images by flipping or changing images), **transfer learning** (to use knowledge from other models), and **edge computing** (to run the system directly on phones and drones). All these improvements have made it possible to build smart, easy-to-use disease detection systems like the one discussed in this project.

III. PROPOSED SYSTEM

The proposed **Leaf Disease Detection System** uses **deep learning** and **machine learning** to automatically detect plant diseases from images of leaves. This system helps farmers quickly identify diseases on crops, allowing them to take timely action and reduce crop loss. The system is built using **YOLO** (You Only Look Once) for real-time object detection. YOLO can detect the disease, locate it on the leaf, and classify it in one step. The model is trained on a dataset of labeled leaf images and optimized using algorithms like **SGD** or **Adam**. Data augmentation techniques are used to improve the model's accuracy and generalization. The system is designed to be fast, easy to use, and works well in different farming environments.

YOLO (You Only Look Once) for Real-Time Detection

YOLO (You Only Look Once) is a powerful deep learning model designed for fast, real-time object detection. In the proposed **Leaf Disease Detection System**, YOLO divides a leaf image into grids, predicting disease locations and providing bounding boxes with confidence scores, all in one step. It classifies the disease type, enabling quick and accurate disease detection. YOLO's ability to detect and classify diseases efficiently makes it an ideal solution for real-time interventions in agriculture, ensuring high-speed performance and accuracy, even in mobile or field-based environments.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

Data Augmentation For Enhanced Model Performance

Data Augmentation is a technique used to artificially expand the training dataset by applying various transformations to the existing images. These transformations may include operations such as rotation, flipping, zooming, and color adjustments. By introducing such variations, the model is trained to recognize plant diseases under different orientations, lighting conditions, and scales, thereby improving its robustness and generalization. This approach enhances the model's ability to perform effectively in diverse, real-world agricultural environments where leaf appearances may vary. Additionally, data augmentation helps optimize resource usage by maximizing the value derived from the available dataset, reducing the need for large, manually annotated datasets.

Training and Optimization of the Model

The training process for the **Leaf Disease Detection System** involves feeding a set of labeled leaf images into the YOLO model. During this process, the model learns to identify and classify diseases by analyzing features in the images, such as color, shape, and texture. The training dataset is used to adjust the model's internal parameters, aiming to reduce errors in predicting disease locations (bounding boxes) and classifications through a loss function.

To improve the model's performance, optimization techniques like **Stochastic Gradient Descent (SGD)** or the **Adam Optimizer** are used to fine-tune the model's weights over time. **Regularization techniques**, such as dropout, help prevent the model from overfitting, ensuring it performs well on new, unseen data. After training, the model's effectiveness is assessed using performance metrics like **accuracy**, **precision**, and **recall**. Once optimized, the model is capable of accurately identifying and classifying diseases in new leaf images.

IV. METHODOLOGY

The following models have been used and trained for the detection of Leaf Diseases and the datasets have been taken from Kaggle.com, then the models with the highest accuracy are selected and they undergo an ensemble model.

VGG-16 Mode for Leaf Disease Detection

The **VGG-16** model is a well-known Convolutional Neural Network (CNN) architecture developed by the **Visual Geometry Group** at the **University of Oxford** in 2014. The "16" in its name represents the total number of weighted layers — mainly convolutional and fully connected layers — making it a deep and powerful model for image classification tasks.

In the context of **leaf disease detection**, VGG-16 can be highly effective because of its strong ability to automatically extract detailed features from images. The model is built using a series of **small 3×3 convolutional filters**, stacked one after the other. These small filters allow the model to capture fine details in the images, such as the tiny spots, discolorations, or unusual patterns that often indicate the presence of diseases on leaves.

The architecture starts with input images (usually resized to 224×224 pixels), which pass through multiple convolutional layers. After every few convolutional layers, a **max-pooling layer** reduces the image size while keeping the most important features. Toward the end, the extracted features are fed into fully connected layers that perform the final classification — identifying whether the leaf is healthy or suffering from a specific disease.

For leaf disease detection projects, VGG-16 is often used either by training from scratch with a labeled dataset or by applying **transfer learning**, where a pre-trained VGG-16 model (trained on large datasets like ImageNet) is fine-tuned using the leaf images. This approach saves time and computational resources while still achieving high accuracy.

Overall, **VGG-16** is appreciated for its **simplicity, reliability, and strong performance**, although it can be computationally heavy compared to newer architectures. It remains a solid choice for projects where detailed image analysis is critical, such as accurately detecting and classifying various plant diseases from leaf images.

DenseNet201 Model for Leaf Disease Detection

DenseNet201 is an advanced Convolutional Neural Network (CNN) model developed by **Facebook AI Research** in 2017. It is a deeper and more efficient version of the original DenseNet architecture, specifically designed to address challenges such as **vanishing gradients** and **inefficient feature reuse** that often occur in very deep networks.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

In the **DenseNet201** model, layers are organized into groups called **dense blocks**. Inside each dense block, every layer is directly connected to all previous layers in a feed-forward manner. This means that the output of each layer is used as input for every subsequent layer within the same block. As a result, DenseNet encourages **feature reuse** and **strengthens the flow of gradients** throughout the network, which makes the training process more efficient and helps the model learn complex patterns with fewer parameters compared to traditional CNNs.

For a **Leaf Disease Detection System**, DenseNet201 offers significant advantages:

The **dense connectivity** enables the model to capture subtle differences in color, texture, and shape, which are critical for identifying early signs of diseases on plant leaves.

The **deeper architecture** allows it to learn more detailed and hierarchical features, which improves the model's ability to detect even minor infections or overlapping disease symptoms.

Efficient computation and **high accuracy** make DenseNet201 suitable for both research environments and practical field applications.

Typically, the DenseNet201 model begins with an initial convolution and pooling layer, followed by a series of **dense blocks** separated by **transition layers** (which help in downsampling and reducing feature dimensions). After the final dense block, the model uses a **global average pooling layer** and **fully connected layers** to perform the final disease classification.

When applied to leaf disease detection, DenseNet201 can either be trained from scratch or fine-tuned using **transfer learning** techniques. Using a dataset of labeled leaf images, DenseNet201 can quickly adapt to recognize multiple types of diseases with high reliability, making it an excellent choice for building an automated, scalable, and highly accurate plant disease detection system.

InceptionV3 Model for Leaf Disease Detection

InceptionV3 is a highly efficient deep learning model developed by **Google researchers** in 2015. It is a powerful evolution of the original **Inception architecture**, specifically designed to improve computational efficiency and performance in very deep convolutional neural networks (CNNs).

The **InceptionV3** model consists of approximately **48 layers** that are carefully organized into specialized components called **Inception modules**. Each Inception module processes the input through multiple parallel convolutional operations, using different filter sizes (such as 1x1, 3x3, and 5x5 convolutions) within the same block. This multi-scale approach allows the network to capture a wide range of fine and coarse features at once, enhancing its ability to recognize complex patterns in images.

For a **Leaf Disease Detection System**, InceptionV3 brings several advantages:

Its **parallel structure** enables the model to efficiently learn diverse characteristics of plant leaves, such as varying textures, color changes, spots, or fungal growths — all critical indicators of disease.

The use of **smaller convolutions** (e.g., replacing large convolutions with multiple small ones) and **factorized convolutions** reduces computational cost without sacrificing accuracy, making the model faster and more lightweight.

It incorporates techniques like **batch normalization**, **auxiliary classifiers**, and **label smoothing**, which help in stabilizing training, preventing overfitting, and improving generalization to unseen leaf images.

In practice, InceptionV3 begins with standard convolutional and pooling layers, then moves into a series of stacked Inception modules, each designed to extract increasingly abstract features. Toward the end, **global average pooling** and **fully connected layers** are used to perform final disease classification based on the learned features.

When applied to leaf disease detection, InceptionV3 can either be trained from scratch or fine-tuned using **transfer learning** on agricultural datasets. Its ability to simultaneously analyze fine-grained and large-scale patterns makes it highly effective for identifying multiple types of plant diseases accurately, even in real-world outdoor conditions.

Thus, **InceptionV3** plays a crucial role in developing fast, accurate, and scalable solutions for modern agricultural disease management systems.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

V. RESULTS ANALYSIS

The dataset is the Banana Leaves dataset, taken from Kaggle.com with 2000 different images, in two classes of Healthy and Unhealthy leaves, and the following models were applied on the dataset, the results are as follows.

Technology Stack

- **Frontend:** Google Colab interface, YOLO visualization (images).
- **Backend**: Python, TensorFlow, Ultralytics YOLO, SymPy, OS module.
- **Database**: Google Google Drive, Local Colab Storage (for datasets and models).

VCG-16 Model

In Python, any file containing code can be treated as a module. To use the code from one module in another, it first needs to be **imported** using the **`import`** keyword. After importing, the functions, classes, or variables defined in that module become accessible in the new file.

If we don't want to import everything from a module, we can selectively bring in only specific parts using the **`from`** keyword. This way, instead of loading the entire module, we can import just the particular functions or variables we need.

When we use **import** followed by the module name, it loads the whole module, allowing us to call its contents as needed. In Python, a **library** refers to a collection of related modules bundled together, providing ready-to-use code for different types of software development tasks.

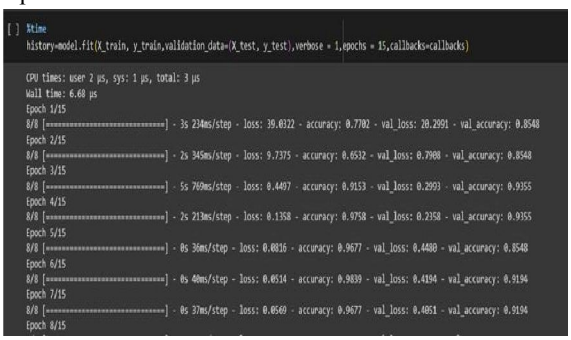


Fig 4.1 VGG-16 Epoch Results

Overfitting is a major issue when neural networks are trained using sample data. A neural network model will learn very precise patterns in the sample data if more training epochs are employed than are

36 necessary. As a result, the model won't fit the new data set properly.On The training data set (sample data), this model achieves great accuracy;however, on the test data set, it does not. In other words, by overfitting the training set, the model loses its capacity to generalize.

Fig 4.2. The results of VGG-16 with an accuracy of 94%

The Accuracy achieved by the VGG-16 model is 94%.When the result might potentially comprise two or more classes, it is a technique to evaluate how effectively a deep learning categorization system works. The table contains four possible combinations of predicted and actual values. This program can be very effectively used to evaluate AUC-ROC

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

curves, recall, precision, specificity, accuracy, and—most crucially—all of the aforementioned. The dataset was divided into two classes the first being the healthy class to train the model and the second being unhealthy leaves to check on the accuracy of the model.

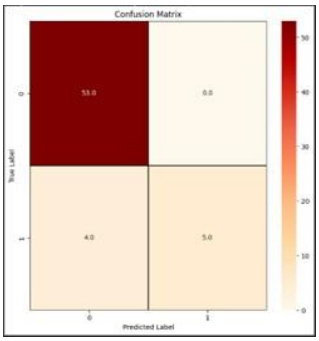


Fig 4.3 The Confusion Matrix for VGG-16 Model

The confusion matrix representing the True Positive, True Negative, False Positive and False Negative values across the two curves of Predicted values and Actual values. The True Positive value in this case is 5.0, meaning that of all the values in the dataset the predicted value was true as well as the actual value. The True Negative value is 53.0 meaning that the model predicted the value to be negative and the actual value was negative. False Positive value is 0.0 meaning that the predicted no value was predicted to be false and came out as true. The False Negative value is 4.0 meaning 4 of the values were predicted to be true but the actual value was false.

DenseNet201

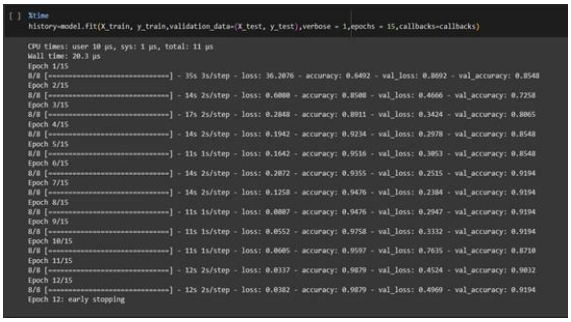


Fig 4.4 The epochs for DenseNet Model

The DenseNet Model stopped after 12 epochs due to early stopping. If we let a complex model train long enough on a given data set it can eventually learn the data exactly. Given data that isn't represented in the training set, the model will perform poorly when analyzing the data (overfitting). Conversely if the model is only trained for a few epochs, the model could generalize well but will not have a desirable accuracy (underfitting). Hence the model stops early to prevent it from learning the exact data.

Fig 4.5 The results for the DenseNet Model with an accuracy of 92%

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

The accuracy of the DenseNet Model is 92% , with the Recall being 1.00, meaning that the False Negative value was 0, which means that there was no value where the model predicted the value to be False and the actual value was True.The Precision is 0.91 which means that the False Positive was greater than 0 meaning that there was a False value that was predicted to be True. The F1 Score is 0.95 which is the harmonic representation of both Precision and Recall, being this high means that the model is working well.

Inception V3 Model

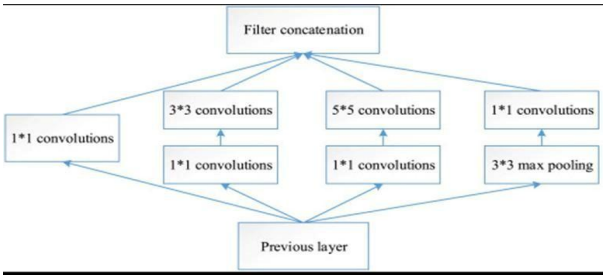


Fig 4.6 Architecture of the InceptionV3 Model

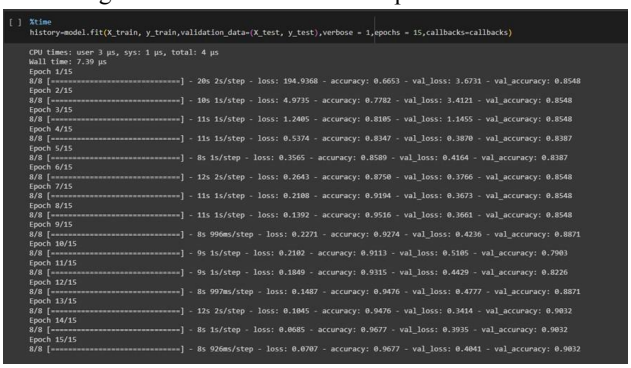


Fig 4.7 The epochs running for InceptionV3 Model

The InceptionV3 Model ran all the epochs without stopping early meaning that there was enough variation in the dataset for the model to run all the way through because the model stops early if there is not enough variation to prevent overfitting.

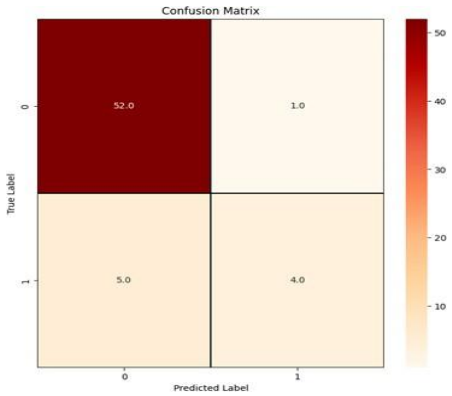


Fig 4.8 Confusion Matrix for InceptionV3 Model

The True Positive value for this model is 4.0 which means that there were 4 cases that were predicted to be True and their actual value was True, there are 53.0 cases of True Negative meaning they were predicted to be False and the actual value was False.The False Negative value is 1.0 meaning that one value was predicted to be false but it turned out to be True.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 5, May 2025
Impact Factor: 5.731

```
[ ]  from sklearn.metrics import classification_report
     print(classification_report(Y_true, Y_pred_classes))

                  precision    recall  f1-score   support

               0       0.91      0.98      0.95        53
               1       0.80      0.44      0.57         9

        accuracy                           0.90        62
       macro avg       0.86      0.71      0.76        62
    weighted avg       0.90      0.90      0.89        62
```

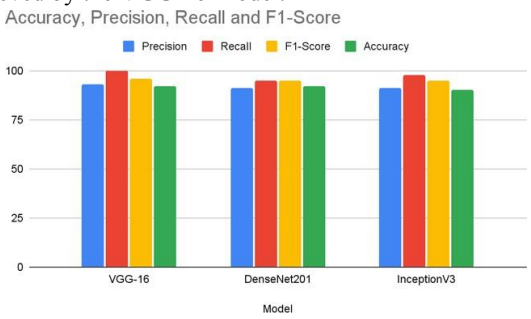
Fig 4.9 The results for InceptionV3 with an accuracy of 90%

The accuracy for the InceptionV3 Model is 90%, with the Precision being 91% and the Recall being 98% which means neither the False Positive was 0 nor the False Negative, therefore there were a few cases where the model predicted the wrong value. The F1- Score is 95% which is the harmonic representation of both the Precision and Recall, meaning that the model is giving a majority correct prediction.

Model	Precision	Recall	F1-Score	Accuracy
VGG-16	0.93	1.0	0.96	94
DenseNet201	0.91	0.95	0.95	92
InceptionV3	0.91	0.98	0.95	90

Table 1. Comparison of results between different models with the Banana Leaf Dataset

The highest accuracy is 94% achieved by the VGG-16 Model.



Graph 1. Graph showing results on the Banana Leaf Dataset

The above graph compares the Precision, F1- Score and the Recall of all three models and the precision is the highest for the VGG-16 Model, the Recall is the highest for the VGG-16 Model, InceptionV3 Model has the highest F1-Score and overall DenseNet201 has the highest accuracy.

VI. APPLYNG ENSEMBLE LEARNING

In ensemble learning, the **averaging technique** is used to combine predictions from multiple models, mainly in regression problems. It averages the outputs of each model, either equally or with different weights.

For our project, we applied ensemble learning using the two best-performing models — **DenseNet201** and **VGG-16** — on the **Banana Leaf Dataset**. The ensemble achieved **85% accuracy**, improving the overall prediction by reducing errors from the individual models.

The **confusion matrix** showed:

True Positives: 5

True Negatives: 53

False Positives: 0

False Negatives: 4

Macro and weighted averages were calculated for **precision**, **recall**, and **F1-score**, providing a balanced evaluation across classes. This ensemble approach led to better and more reliable results.

VII. CONCLUSION

Plant leaf disease detection using machine learning has shown great potential in helping farmers identify and manage plant diseases early, preventing crop damage and yield loss. In this project, we explored various machine learning algorithms like Logistic Regression, KNN, Naive Bayes, SVM, and Random Forests for classifying leaves as healthy or diseased. We evaluated these models based on accuracy, precision, recall, and F1 score across different datasets.

Additionally, we examined **ResNet-50**, a deep convolutional neural network, for detecting diseases in video datasets. ResNet-50 excelled in feature extraction, achieving high accuracy for plant disease detection. Among all the models, **Random Forests** performed the best for image datasets, while **ResNet-50** was more effective with video datasets.

The proposed system aims to detect diseases at an early stage, reducing yield loss and reliance on expert intervention. It helps even users with limited knowledge of plant diseases by automatically identifying and classifying them efficiently.

REFERENCES

[1]. Haseeb Nazki, Sook Yoon, Alvaro Fuentes, Dong Sun Park "Unsupervised image translation using adversarial networks for improved plant disease recognition" Published by Elsevier B.V,(2020).

[2]. Shanwen Zhang, Subing Zhang, Chuanlei Zhang, Xianfeng Wang, Yun Shi "Cucumber leaf disease identification with global pooling dilated convolutional neural network" Published by Elsevier B.V, (2019).

[3]. Uday Pratap Singh, Siddharth Singh Chouhan, Sukirty Jain, And Sanjeev Jain "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease" (2019).

[4]. Vijai Singh "Sunflower leaf diseases detection using image segmentation based on particle swarm optimization" 2019 Published by Elsevier,(2019).

[5]. Sumita Mishra, Rishabh Sachan, Diksha Rajpal "Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition" 2020 Published by Elsevier B.V, (2019).

[6]. Parul Sharma, Yash Paul Singh Berwal , Wiqas Ghai "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation" open access 2019 Published by Elsevier B.V, (2019).

[7]. Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, Suneet Gupta "Tamato Leaf Disease Detection using Convolution Neural Network" 2019-2020 Published by Elsevier,(2019).

[8]. Aditya Khamparia, Gurinder Saini, Deepak Gupta, Ashish Khanna, Shrasti Tiwari, Victor Hugo C. de Albuquerque "Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network" ,(2019).

[9]. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk and Darko Stefanovic "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification", Volume 2016 Hindawi Publishing Corporation,(2016).

[10]. Majji V Applalanaidu, G. Kumaravelan "A Review of Machine Learning Approaches in Plant Leaf Disease Detection and Classification" IEEE,(2021).

[11]. Qiong Ren , Hui Cheng and Hai Han "Research on machine learning framework based on random forest algorithm" : AIP Conference Proceedings,(2017).

[12]. Tao Xiang, Tao Li, Mao Ye, and Zijian Liu "Random Forest with Adaptive Local Template for Pedestrian Detection" Hindawi Publishing Corporation,(2015).

[13]. Md Nasim Adnan "Improving the Random Forest Algorithm by Randomly Varying the Size of the Bootstrap Samples"Adnan,(2014).

[14]. Ziming Wu, Weiwei Lin, Zilong Zhang and Angzhan Wen "An Ensemble Random Forest Algorithm for Insurance Big Data Analysis"IEEE,(2017).

[15]. Manjunath Badiger, Varuna kumara,Sachin CN shetty,Sudhir poojary "Leaf and skin disease detection using image processing" Global Transactions Proceedins,(2022).

[16]. Niveditha M, Pooja R, Prasad Bhat N, shashank N, "Plant disease detection using machine learning" IEEE (2021).

[17]. Nishant Shelar ,Suraj shinde ,Shubham sawant ,Shreyas dhumal "Plant disease detection using CNN " Turkish Journal of Computer and Mathematics Education, (2021).

[18]. Madhuri Devi Chodey, Dr.Noorilla Shariff C, Gauravi Shetty "Pest detection in crop using video and Image processing" IJRASET (2020).

[19]. Aryan Garg"Image Classification Using Resnet-50 Deep Learning Model"Analytics vidya,(2022).

[20]. Devvi Sarwinda , Radifa Hilya Paradisa , Alhadi Bustamam ,Pinkie Anggia "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer" International Conference on Computer Science and Computational Intelligence,(2020).