"Classification of leaf disease Using Deep Convolutional Neural Networks"

Abstract—**This paper proposes a novel approach to improving leaf disease classification using convolutional neural networks (CNNs) and the SoftMax activation function. The proposed approach extracts features from images of the leaf and hands using a CNN and then classifies the leaf disease into one of several categories using a SoftMax activation function. The proposed approach was evaluated on a dataset of images of leaf collected, and the results showed that it was able to accurately classify leaf disease with a good accuracy of over percentage. The SoftMax activation function plays a crucial role in the proposed approach, as it allows the CNN to output a probability distribution over the different classification leaf disease categories. This probability distribution can then be used to identify the most likely leaf disease category and to provide feedback to the leaf. The proposed approach has the potential to be used to develop leaf monitoring systems that can help to improve crop production quality. For example, a rice leaf monitoring system could use the proposed approach to detect when a leaf is unhealthy or faded, and Farmer identify the leaf disease.**

Index Terms—**Leaf Disease Classification, Deep Learning, CNN, Image Classification**

I. Introduction

AGRICULTURAL productivity, the foundation of global food security, is perennially challenged by the widespread threat of leaf diseases. Plant leaf health is an important indicator of overall crop health, and diseases affecting these vital organs can cause substantial yield losses. In recent years, the integration of advanced technologies, particularly in the areas of machine learning and computer vision, has emerged as a promising way to revolutionize leaf disease detection and management. The literature on leaf disease classification is replete with studies and advances that demonstrate the evolution of methods and technologies employed in this complex domain. As agriculture has become increasingly data-driven and technology-centric, the need for accurate, scalable and efficient disease classification systems has intensified. The aim of this literature review is to provide a comprehensive overview of key issues, methods and classification of leaf diseases.

An exemplary technique that has significantly improved leaf disease classification is the use of convolutional neural networks (CNNs). CNNs have proven to be a game-changer in image-based classification tasks, providing a profound ability to automatically learn complex features from raw data. This technology, originally developed for image classification in various domains, has found wide application in agriculture, especially in leaf disease detection.

The purpose of the Leaf Diseases Classification project is to deploy machine learning and computer vision to enable early detection and precise identification of leaf diseases in plants. By providing farmers with a tool for timely intervention and resource optimization, the project aims to enhance crop health, support sustainable 1 agricultural practices, and contribute to global food security.

In agriculture, the threat of leaf diseases poses a constant threat to crop health and yield. Traditional manual inspection methods are time-consuming and error-prone. To address this challenge, our project focuses on applying advanced machine learning techniques for intelligent leaf disease classification. Using deep learning, specifically Convolutional Neural Networks (CNN), we aim to develop a robust system capable of quickly and accurately detecting various leaf diseases. The project promises to revolutionize crop management, providing farmers with a reliable, automated tool for early disease detection and intervention. The following sections will elaborate on the challenges, methods, and expected benefits of this innovative leaf disease classification system.

- Early Disease Detection
- Increased Crop Yield
- Minimized Economic Losses
- Optimized Resource Utilization
- User-Friendly Interface for Farmers
- Scientific Advancements in Agriculture

II. Literature Study

Deep convolutional neural networks (CNNs) have emerged as a promising tool for identifying leaf diseases. CNNs, a type of machine learning algorithm, excel at pattern recognition in images, making them well-suited for classifying various leaf diseases. Studies have demonstrated their effectiveness in distinguishing between diseases affecting apple, orange, cherry, peach, and potato crops, among others. This technology offers a valuable solution for a range of applications, including leaf disease detection. This review will explore the application of deep CNNs in leaf disease detection, while also addressing the associated challenges and limitations.

In [1] The focus of this research area lies on leveraging deep learning, specifically CNN-based models, for leaf disease classification. These models hold the potential to surpass human accuracy in disease identification, exceeding 94 percent accuracy according to some studies. However, a key challenge is the model's dependence on the training data. When tested with images from a different dataset than the one used for training, accuracy can plummet, as evidenced by a drop from 99.01 percent to 45.95 percent in one study.

In [2] this paper mentions the use of machine learning methods such as support vector machine (SVM), K-means clustering, deep learning, and K-nearest neighbors (K-NN) for disease identification and classification. Also, FLDA gave an accuracy of 90 percent for heavily damaged leaves. The challenge is to use these methods on the research gaps that are observed and still exist in research on Plant disease detection and classification Automatic cluster center initialization is lacking, improves the proposed algorithm for reduction of error due to classification, To integrates advanced 8 imaging technique and Computer vision algorithms, For automatically detection of plant leaf disease algorithms are still required.

In [3] A Dataset for Visual Plant Disease Detection" includes MobileNets Object Detection Network, SSD framework, and Faster R-CNN with InceptionResnetV2. Our results show that modeling using our dataset can increase the classification accuracy by up to 31 percent. Further, to train highly accurate models for disease detection, we may require a dataset with more number of images in each class. However, the approach gives a feasible direction to tackle the ongoing problem of disease detection.

In [4] the paper Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification, Najdenovska et al. used plant, automated classification of the plant's abnormal state caused by spider mites, and this study got an accuracy of 80 percent.

In [5] A Review of Machine Learning Approaches in Plant Leaf Disease Detection and Classification as inceptionv3, ResNet, and

XceptionNet have great depth with fewer parameters. They have achieved the classification accuracy of 99.76percentage, 96percentage, and 98.7percentage respectively. The least number of parameters in GoogLeNet, with more layers showed 99.35percentage classification.

In[6] This paper uses a deep Convolutional Neural Network (DCNN) with Bayesian learning, achieving an accuracy rate of 98.9 percent without over fitting, utilizing CNN features in different hierarchies.

In [7] this paper Plant leaf disease classification using deep Convolutional neural network with Bayesian learning, the deep learning approach improves significantly by 40–42percentage average. The result shows that machine learning does not improve the classification of leaf disease because of the following reasons: Features are not efficient in describing affected leaf parts. Non-linear mapping of features not present. Only work on low-level features. 15 classes increase the overlapping and 9 reduce the performance.The proposed network can detect the type of disease with an accuracy of 98.9 percent with no sign of overfitting.

In [8] The proposed model is Deep learning and transfer learning techniques, specifically the Efficient Net deep learning architecture, for the classification of plant leaf diseases. The B5 model achieved 99.91percentage accuracy and 98.42percentage precision in the original dataset, while the B4 model achieved 99.97percentage accuracy and 99.39percentage precision in the augmented dataset.

In [9] The research aims to detect and classify plant leaf diseases using image processing and deep learning techniques utilizes image processing and deep learning techniques, including CNN and SVM, for the detection and classification of plant leaf diseases. focuses on using image processing and deep learning techniques, specifically CNN, to detect and classify plant leaf diseases with an accuracy of approximately 90 percent. A greater and more varied data activation for training the learning model, as well as the accuracy of the disease.

In [10] artificial intelligence techniques for apple leaf disease detection and classification in precision agriculture use of computer vision and artificial intelligence (AI) techniques, including CapsNet and BiLSTM models, for apple leaf disease detection and classification. It highlights the needs and challenges of plant disease detection and diagnosis in precision agriculture.

In [11] The purpose of the research is to discuss the application of deep learning methods. CNN-based classification models, challenges, and solutions in using deep learning for plant disease detection. Artificial Intelligence Enabled Apple Leaf Disease Classification for Precision Agriculture The experimental outcomes demonstrate the promising performance of the AIE-ALDC technique over the recent state-oftheart methods with a higher accuracy of 0.9920. Need diverse and representative datasets. accurate disease recognition and detection are highlighted as significant 10 challenges to visual inspection and the complexity of disease characteristics.

In [12] He researches the purpose is Convolutional neural networks (CNNs), The dataset, PlantDoc, image processing techniques, MobileNets Object Detection Network, feature extraction, and Neural Network Ensemble (NNE). performance of the AIE-ALDC technique over the recent state of art methods with a higher accuracy of 0.9920. Ack of availability of sufficiently large-scale non-lab datasets. lack of expertise, lab infrastructure for disease identification challenge of timely disease detection.

In [13]The purpose of this research the study aims to focus on three well-known rice diseases, namely, leaf smut, brown spot caused by fungus, and bacterial leaf blight caused by bacteria Convolutional networks (CNNs). Real-Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique This method achieved a 99.17 percentage validation and 98.57 percentage test accuracy. It aims to fill the research gap in the automated identification and classification of rice diseases using deep learning techniques.

In [14] The research aims to improve disease detection in grape plants using a convolutional capsule network, a type of deep learning method, to achieve superior results compared to existing methods. it's a final model with 99.7 percent accuracy. Convolutional neural networks (CNN). The proposed method aims to overcome these challenges and contribute to the advancement of disease detection in agriculture.

Research on leaf disease classification using machine learning algorithms. This literature review provides a comprehensive analysis of the state-of-the-art in leaf disease classification with a primary emphasis on image processing techniques. The integration of computer vision and machine learning into disease identification processes stands out as a transformative approach, offering innovative solutions to longstanding challenges in agriculture and plant pathology

III. Materials and Methods

Detecting leaf diseases on time is crucial for crop health and yield utilization in agriculture. The agricultural sector plays a significant role in the feasible of global food security. This project, named Leaf classification, aims to support the power of deep learning techniques to create a strong and efficient model for the automated classification of leaf diseases. Classification of leaf disease proposes a deep learning model that leverages convolutional neural network (CNN) and we are testing to machine few pre-training model as VGG16, ResNet50, GoogleNet, Mobilenet50.We went to recognize the device using 70295 image for leaf , we can recognize it by seen at it from different angles, and we will use this image for our train developed model. Here we use the SoftMax activation function can detect and classification the image.

Dataset: Dataset is a very important part of every project. The accuracy of the model relies on the perfectness of the dataset. The purer the dataset, the more accuracy. Various methodologies have been discussed and applied by many researchers. This paper needs a dataset of leaf disease classification and condition. The data used in this study came from the Kaggle website and was used to demonstrate the results of deep learning algorithms implemented in Python. We use the SoftMax activation function, so we need image data for training the model. Because the more data we train, the more accurate results we gain.

Number of Total Images	70295
Number of Classes	38
Number of Train Images	56236
Number of Validation Images	14059

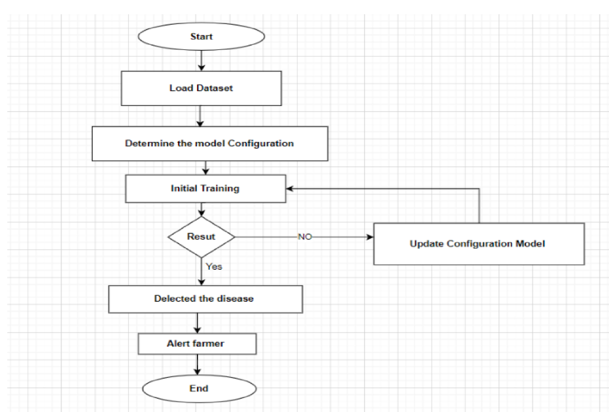

Fig. 1: Outline of the assistance system

A. *Data Analysis and Pre-processing:*

Here we prepare our leaf data for training and evaluation. This typically involves splitting the data into training and testing sets, scaling the data, and creating sequences or windows of input-output pairs for the models. As we know our dataset is an image series dataset. For better results, we have done pre-processing. Preprocessing is a crucial step in data analysis and modeling tasks, including leaf classification and leaf health condition. It involves transforming, cleaning, and preparing raw data before feeding it into a model or algorithm. Pre-processing includes selecting the most relevant and informative features (variables) for leaf disease classification. This step's helpful attributes that may introduce healthy and unhealthy leaves or leaf spots impact the model's performance. By selecting the right set of features, preprocessing enhances prediction accuracy and efficiency.

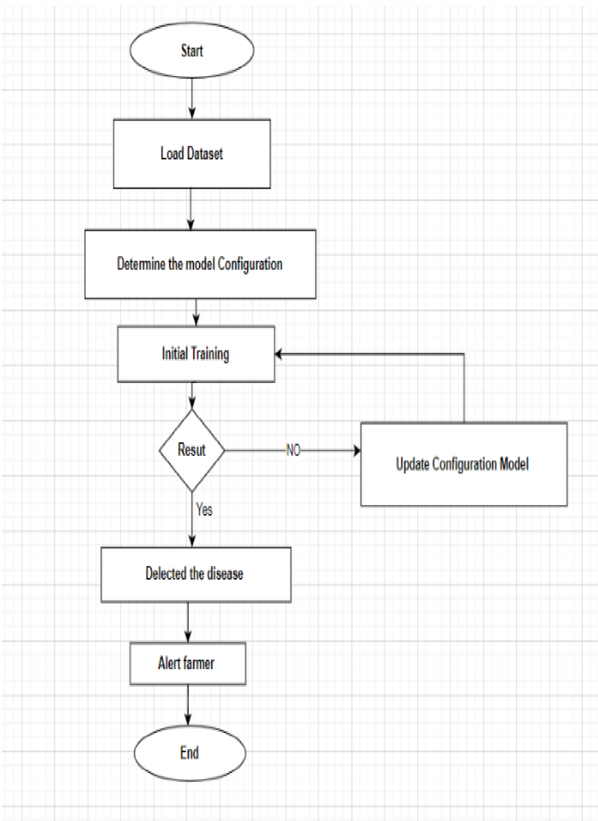

Fig. 2: Data analysis data flow diagram

B. Proposed Model

In neural network classifiers, the SoftMax activation function is frequently employed in the final layer. It transforms the network's outputs, typically real numbers, into a probability distribution across the possible output classes. This ensures that the sum of the SoftMax function's outputs equals one, where each output represents the probability of the input belonging to a specific class.
Our dataset will consist of 70295 images categorized into 38 distinct classes, which may further contain subclasses. Through a testing phase, we aim to train the device on these images to achieve functional performance. As an example, we could utilize an image of a tomato leaf. This information, including the image itself, would be incorporated into our training data. Our project strives for high accuracy.
We propose utilizing a convolutional neural network (CNN) model, potentially leveraging pre-trained models such as GoogleNet, VGG16, ResNet50, or MobileNet50. Pre-trained models are saved networks that have undergone prior training on extensive datasets, often focused on large-scale image classification tasks.

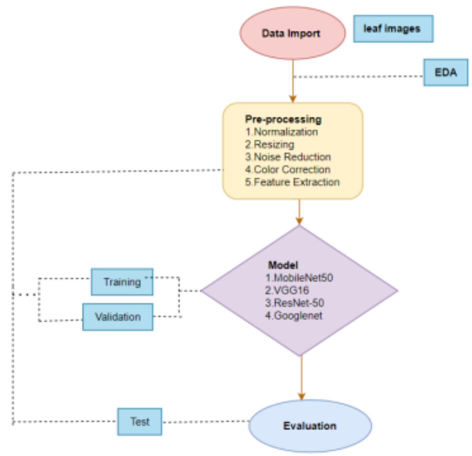

Fig. 3: Block diagram of the proposed methodology

We will use a total of 70295 images as input. We will divide into 38 classes and these have some subclasses. We will assume that we will do some testing to train the device on the images and get some working ones. Suppose we take a tomato leaf, this information and picture is ours. It is kept in the class and then I will test it. We are hoping for good accuracy in our project We are used tosomepretrained GoogleNet, VGG16, ResNet50, and MobileNet50 We proposed a convolutional neural network(CNN) model. A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. we are used to some pre-trained GoogleNet, VGG16, ResNet50, MobileNet50

C. Model Development

In the context of leaf disease classification, the ResNet50 (Residual Network) architecture is a robust and popular choice due to its depth, accuracy, and ease of training. Developed by Microsoft Research. Below is an overview of the ResNet50 architecture and its application to leaf disease classification.

D. Overview of ResNet Architecture

ResNet is a deep convolutional neural network (CNN) architecture designed to address the challenges of training very deep networks. It introduced the concept of residual blocks.

Residual Blocks At the core of ResNet50 are residual blocks. These blocks include a series of convolutional layers with a shortcut connection that skips over these layers. The output of the block is the sum of the input and the processed layers' output. This design helps address the problem of vanishing gradients and enables training of much deeper networks. Architecture Variants: ResNet50 comes in several variants with different depths, typically denoted by their layer counts. The deeper the network, the more residual blocks it contains.

Layers in ResNet50: ResNet50 typically consists of the following key layers/components.

Convolutional Layers: These layers perform convolutions, applying filters to the input to extract features.

Batch Normalization: Each convolutional layer is usually followed by batch normalization, which helps stabilize training by normalizing the inputs to each layer.

Activation Functions: Typically, ReLU (Rectified Linear Unit) is used to introduce non-linearity.

Pooling Layers: Max-pooling is often used to reduce spatial dimensions and focus on more significant features.

Fully Connected Layers: After a series of residual blocks, the network typically concludes with one or more fully connected layers that output the classification results.

Applying ResNet50 to Leaf Disease Classification: In a leaf disease classification project, the ResNet50 architecture can be used as follows.

Preprocessing and Data Augmentation: Before feeding data into ResNet50, preprocessing techniques (resizing, normalization) are applied. Data augmentation, like rotations, flips, and brightness adjustments, can improve robustness.

Feature Extraction and Transfer Learning: Given that ResNet50 is a pretrained model on large datasets like ImageNet, transfer learning can be employed. By using a pre-trained ResNet50 and fine-tuning it with leaf disease datasets, one can use pre-learned features while customizing for leaf-specific diseases.

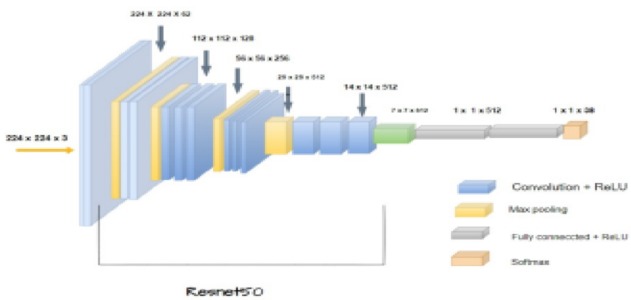

Fig. 4: Resnet50 model diagram

Residual Blocks for Feature Learning: ResNet50's residual blocks facilitate deeper feature extraction, allowing the network to learn complex patterns and identify disease-related features in leaf images.

Final Classification: The final output layer typically uses a softmax activation function to output probabilities for each class (disease type). The network is trained to minimize a loss function, such as categorical cross-entropy, to achieve accurate classification.

E. Data Collection and Preprocessing

Data Collection: We collected healthy and unhealthy images for our model. These images were labeled according to the disease type.

Data Augmentation: We applied techniques such as rotation, flipping, and color adjustments to increase the dataset's diversity and prevent overfitting.

Data Splitting: The dataset was divided into training, validation, and test sets to ensure that the model generalizes well.

F. Model Configuration

Base Model Selection: A pre-trained ResNet50 model, such as ResNet-50, was chosen as the base model. Using a pre-trained model allows for transfer learning, where the model is fine-tuned with the specific dataset for leaf disease classification.

Model Architecture: The base ResNet50 model was modified to include custom layers for leaf disease classification. This typically involves replacing the final fully connected layers with ones specific to the number of classes in the leaf disease dataset.

Loss Function and Optimizer: The cross-entropy loss function was used to measure the accuracy of predictions. An optimizer, such as Adam, was selected for training the model, ensuring convergence and efficient training.

Training and Evaluation: Training: The model was trained on the training set, with validation against the validation set to monitor performance and prevent overfitting.

Evaluation: After training, the model was tested on the test set to evaluate its accuracy and generalization capabilities.

Model Performance: Accuracy and Loss: Accuracy and loss metrics were used to evaluate the model's performance during training and testing. Confusion Matrix: A confusion matrix was generated to visualize the classification results and identify potential areas for improvement.

Fine-Tuning and Hyperparameter Tuning: If necessary, hyperparameters such as learning rate and batch size were adjusted to improve the model's performance.

Algorithm: ResNet50 addresses the challenges of deep networks by introducing "residual learning." The core idea is to use shortcut connections, also known as skip connections, which allow gradients to flow more easily through the network. The structure of ResNet50 is built around "residual blocks."

Residual Blocks: A residual block consists of one or more convolutional layers, with a shortcut connection that skips these layers. The output of the residual block is the sum of the input and the output from the convolutional layers. This design allows information to flow through the shortcut, avoiding degradation in deeper networks. Mathematically, a residual block can be represented as:

$$y = f(x) + x$$

where x is the input to the residual block, $f(x)$ is the transformation applied by the convolutional layers, and y is the output. The addition operation ensures that the shortcut connection retains information, reducing the impact of vanishing gradients.

The following outlines the ResNet50-based algorithm for leaf disease classification:

Data Collection: We collected healthy and unhealthy images for our model. These images were labeled according to the disease type.

Data Augmentation: We applied techniques such as rotation, flipping, and color adjustments to increase the dataset's diversity and prevent overfitting.

Data Splitting: The datase twas divided into training, validation, and test sets to ensure that the model generalizes well.

G. Model Evaluation

Following the training phase, the model's performance should be evaluated on a held-out test set to assess its accuracy and generalization capabilities. Common evaluation metrics employed for this purpose include accuracy, precision, recall, and the F1 score.

IV. Implementation and Testing

Leaf disease classification is a crucial task in agriculture to ensure the early detection and management of plant diseases, which can significantly impact crop yield and quality. In this project, we aim to develop a machine-learning model that can accurately classify various types of leaf diseases, such as fungal, bacterial, and viral infections, based on images of the affected leaves.

The implementation, testing, and analysis of the leaf disease classification model are discussed in detail in this section. This section presents an extensive analysis of the findings. We look at the accuracy, precision, recall, and other significant parameters of the model.

A. Dataset

This dataset was recreated using offline augmentation from the original dataset. The original dataset can be found in the GitHub repo. This dataset consists of approximately 70K RGB images of healthy and diseased crop leaves, which are categorized into 38 different classes. The total dataset was divided into an 80/20 ratio of training and validation sets, preserving the directory structure.

The main phases of this study are shown in the dataflow diagram in Figure 4.2. Gathering data is the first step. Next, we preprocessed the data. After that, the dataset was divided into three groups: validation, testing, and training. Next, we decide on our suggested model. We made an effort to divide leaf disease into thirtyeight classes. Following that, we used a range of measures to assess performance.

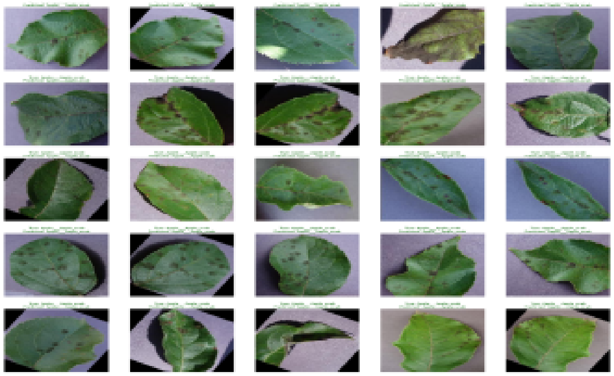

Fig. 5: Samples of Training Dataset image

B. System set up

In the processes of data pre-processing, experimentation, and model evaluations, Python is the designated programming language. Its adaptability and extensive library support make it a preferred choice for tasks involving data manipulation, analysis, and the development of machine learning models. For the implementation of the suggested architecture, TensorFlow and Keras are employed. TensorFlow, an open-source deep learning framework, provides low-level operations for neural network development. Keras, a high-level neural networks API, complements TensorFlow with an intuitive interface, facilitating model construction and experimentation. NumPy plays a pivotal role in mathematical operations within the architecture. As a fundamental library for numerical computing, NumPy supports large, multi-dimensional arrays and matrices, streamlining mathematical computations. Python's versatility is particularly advantageous during experimentation and prototyping, allowing for quick iteration and testing of different architecture components. Libraries like OpenCV and scikit-learn enhance Python's capabilities in data pre-processing, supporting tasks such as image augmentation, normalization,

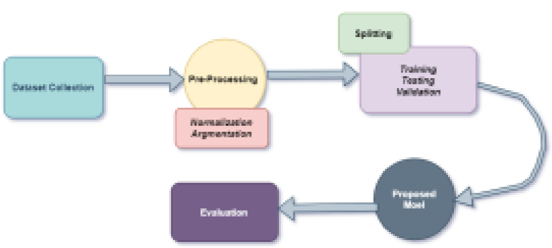

Fig. 6: Data Flow Diagram

and data splitting. During model evaluations, Python's extensive machine learning metric libraries, including scikit-learn, enable the assessment of model performance. Metrics such as accuracy, precision, recall, and ROC Curve can be easily computed and analyzed within the Python environment. In summary, the combination of Python, TensorFlow, Keras, and NumPy forms a cohesive and effective environment for the implementation, experimentation, and evaluation of the suggested leaf Disease classification architecture.

C. Evaluation

To have a thorough grasp of our leaf disease classification model's efficacy, evaluation metrics are essential for evaluating its performance. Assessment metrics function as standards that enable us to gauge and express the degree of performance that our model is achieving. They offer insightful information on its advantages and possible development opportunities. We evaluated our model using the following metrics: Accuracy, ROC Curve, Precision, and Recall. Resnet50 Accuracy: The ratio of accurately predicted occurrences to all instances in the dataset is known as accuracy. It offers a broad picture of a model's performance over all classes or categories.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

When the accuracy is high—nearly 94 percent—it means that the model is predicting most cases in the dataset correctly. It suggests that the model does a good job of generalizing patterns and encapsulating the data's fundamental structure. On the other hand, a low accuracy suggests that the model is having trouble producing accurate predictions. This could be the result of problems like a lack of training data or a complicated challenge. When the number of instances in various classes is significantly out of balance, accuracy might not be the best metric to use. However, accuracy may be the best statistic in situations where the total number of instances is distributed evenly throughout the classes.

Resnet50 Precision : The formal definition of precision is the ratio of genuine positives to the total of false positives and true positives. It is calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Precision is a key evaluation parameter that sheds light on a model's accuracy of positive predictions. In particular, it calculates the percentage of accurately anticipated positive cases, or true positive predictions, out of all positively predicted instances.

When minimizing false positives is crucial, precision becomes especially important. A high precision number means that the model's positive predictions are made with accuracy. It implies that something is very likely to be accurate when the model indicates that it belongs to a particular class. Positive predictions from the model are more reliable when they are made with high precision.

On the other hand, a low precision rating suggests that there are a lot of erroneous positive predictions being made by the model. This implies that the model may not always be accurate when it makes a favorable prediction. When making decisions in situations where false positives have practical repercussions, precision becomes crucial. Precision becomes more revealing than accuracy in datasets where the classes are uneven, i.e., one class considerably outnumbers the other. This is because precision focuses only on the accuracy of positive predictions, whereas accuracy can be deceiving in unbalanced datasets.

The formal definition of recall is the ratio of true positives to the total of false negatives and true positives. It is calculated using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

Resnet50 Recall: Recall is an essential evaluation statistic in classification and machine learning applications. It is sometimes referred to as sensitivity or true positive rate. In particular, it assesses how well a model can recall or capture each positive occurrence in a dataset.

When missing positive occurrences (false negatives) are a major concern, recall becomes even more crucial. A high recall value suggests that most positive cases are well captured by the model. It implies that the model is capable of identifying positive situations and reducing the likelihood of overlooking real positive occurrences.

On the other hand, a low recall value suggests that a sizable portion of positive examples are being missed by the model. This could be a problem in situations where false negatives have significant repercussions, like in medical diagnosis, when a failure to diagnose a condition may result in treatment being delayed.

Recall becomes a more relevant metric than accuracy in datasets where the classes are unbalanced, meaning one class considerably outnumbers the other. This is because recall focuses exclusively on the capacity to catch positive instances, whereas accuracy might be deceptive in unbalanced datasets.

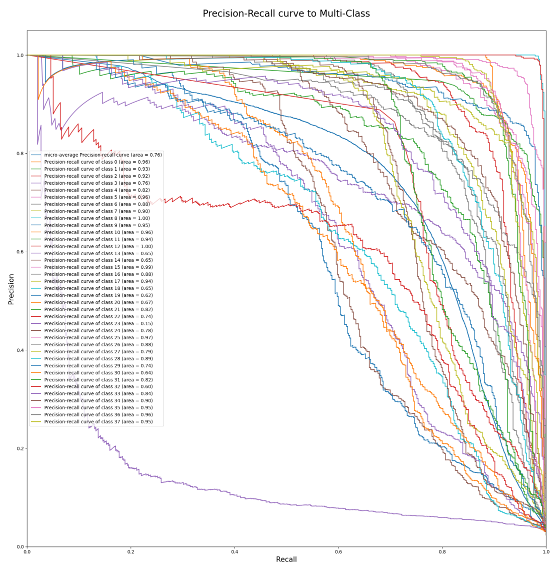

Fig. 7: Precision-Recall curve to multi-class

Resnet50 ROC Curve : A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate (TPR)
- False Positive Rate (FPR)

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{5}$$

False Positive Rate (FPR) is defined as follows:

$$\text{FPR} = \frac{FP}{FP + TN} \tag{6}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

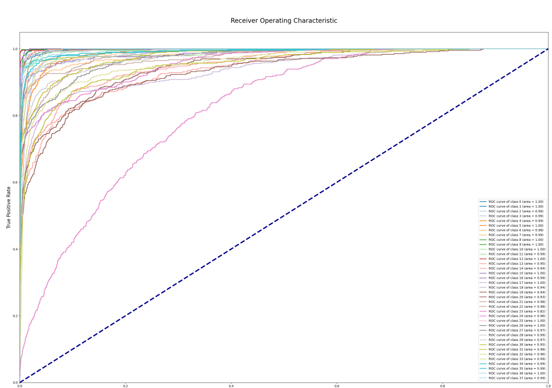

Fig. 8: ROC false positive rate and true positive rate graph.

This graph presents the Receiver Operating Characteristic (ROC) curves for a multi-class classification problem, with each curve corresponding to a different class in our dataset. The x-axis represents the false positive rate, while the y-axis represents the true positive rate. For example, classes 0, 5, 6, 10, and several others have an AUC of 1.0, indicating perfect classification performance. Most classes have AUC values very close to 1.0, suggesting our model performs exceptionally well.

D. Result Evaluation

We conducted training using (10) epochs and a batch size of (32) . On the(10) th epoch, the training accuracy was (94 percent) and validation accuracy was (96 percent.) Our main model is ResNet50. its performance after training and testing in terms of accuracy

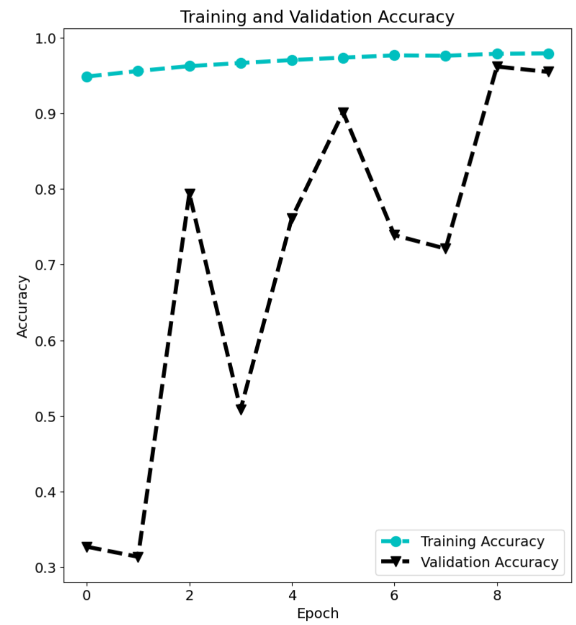

Fig. 9: Training accuracy vs. validation accuracy curve

Training accuracy vs. validation accuracy curve: A machine learning model's performance during training and validation can be seen in the Figure 4.5.1 graphically in the training accuracy vs. validation accuracy curve. It displays the evolution of the model's accuracy over several training iterations or epochs. The number of epochs is represented by the x-axis, and the model's accuracy is represented by the y-axis. Two lines make up the curve: one represents training accuracy and the other represents validation accuracy. The model's performance over time on the training set of data is shown by the training accuracy line. The training accuracy is low at first because the model hasn't learned anything. As the model keeps learning and adjusting its parameters in response to the training data, the training accuracy gets better and better over time. It's important to keep in mind that training accuracy alone does not provide a complete picture of the model's functionality. The model could be deceptive if it simply memorizes the training data and is unable to generalize to fresh, unknown input. The validation accuracy line displays the model's performance on an alternative validation dataset. Using the validation dataset, the model's performance on data that it hasn't seen during training is evaluated. As the model finds broad patterns in the training data, the accuracy of its initial validation improves. However, the validation accuracy may eventually reach a plateau or even start to decrease. This implies that the model is overfitting, which indicates that it is struggling to generalize to new data and has grown unduly specialized to the training set. Overfitting is a common problem in machine learning that the validation accuracy curve helps identify. Figure. 4.6 shows that the training accuracy steadily increases with each epoch, indicating that the model is learning from the training data and improving its performance. The validation accuracy also shows improvement but might exhibit fluctuations. It generally follows the trend of the training accuracy, but it's slightly lower. It's important to monitor the training and validation curves for signs of overfitting. If the training accuracy is significantly higher than the validation accuracy, the model may be overfitting to the training data. The model seems to generalize well, as both training and validation accuracy are increasing and converging.

E. Results and discussion

Epoch	Loss	Accuracy	Val loss	Time
1	0.1542	0.9486	6.641	441s
2	0.1311	0.9559	7.1539	439s
3	0.1140	0.9623	0.8366	437s
4	0.1001	0.9665	4.4763	441s
5	0.0880	0.9704	1.1899	434s
6	0.0805	0.9736	0.4066	438s
7	0.0714	0.9766	1.6969	436s
8	0.0681	0.9761	1.5610	437s
9	0.0647	0.7647	0.1292	436s
10	0.0606	0.6706	0.1525	433s

TABLE I: Result of epoch ResNet50

Epoch	Loss	Accuracy	Val loss	Time
1	2.1255	0.3639	2.3440	457s
2	0.7627	0.7658	1.6333	423s
3	0.5119	0.8474	3.8570	421s
4	0.4008	0.8826	2.0916	425s
5	0.3369	0.9426	0.4240	425s

TABLE II: Result of epoch GoogleNet

Epoch	Loss	Accuracy	Val loss	Time
1	0.4018	0.8708	1.7373	527s
2	0.2811	0.9091	0.7970	510s
3	0.2241	0.9268	1.3418	462s
4	0.1894	0.9375	0.8918	511s

TABLE III: Result of epoch MobileNet50

Epoch	Loss	Accuracy	Val loss	Time
1	0.7587	0.7730	0.3561	1025s
2	0.4407	0.8538	0.3317	1015s
3	0.3846	0.8730	0.3078	1018s
4	0.3496	0.8849	0.2456	1005s
5	0.3310	0.8894	0.2330	1011s

TABLE IV: Result of epoch VGG16

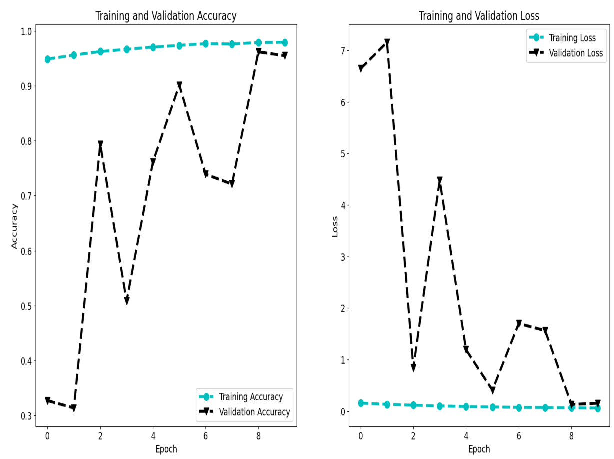

Fig. 10: The training and validation, training validation loss curve epoch= 10 ResNet.

Here we can see brief analysis of the results generated by our model. On the lefthand side, we have the ”Training and Validation Accuracy” graph. The x-axis represents the number of epochs we had run. The y-axis represents accuracy. As we can see, the training accuracy improves rapidly, However, the validation accuracy shows a more fluctuating pattern. The 35 x-axis again represents the epochs, while the y-axis represents the loss.The cyan line shows the training loss, and the black dashed line shows the validation loss.Here, we observe that the training loss decreases steadily, which is expected as the model learns.The validation loss, however, decreases initially but then starts to increase and fluctuate. Due to running short number of epochs we have face some overfitting issue that we can solve in those graph but overall our performance of our model gives satisfactory result.

Authors	Model	Accuracy
Proposed Model	Oure Model	94.97 percent
Aanis Ahmad [1]	deep CNN	99.35 percent
Vijai Singh[2]	DenseNet (Dense Convolutional Network)	99.35 percent
Davinder [3]	Convolutional Neural Network (CNN),B4,B5	96.03 percent
Jinzhu Lu. [4]	CNN	99.0.7 percent
MAJJI V . [4]	GoogLeNet with more layers	99.35 percent
Preeti Singh.2021[5]	CNN	92.6 percent

TABLE V: Comparison of our proposed model with some existing models

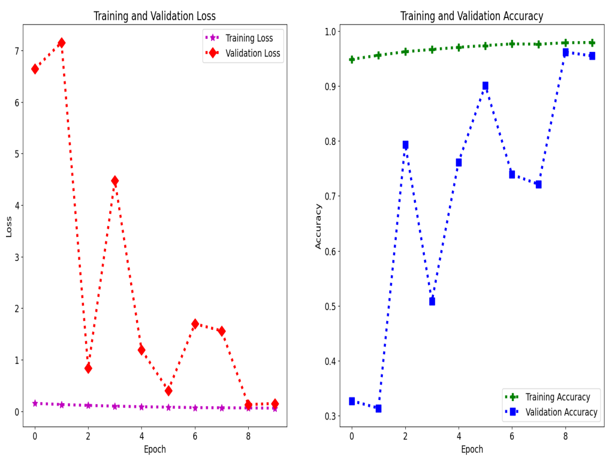

Fig. 11: Comparison of our proposed model with some existing models.

	Resnet50	Googlenet	Mobilenet50	VGG16
Accuracy	0.94	0.72	0.77	0.72
ROC Curve	0.99	0.97	0.99	0.66
Precision	0.90	0.82	0.62	0.61
Recall	0.99	0.76	0.85	0.58

TABLE VI: Comparison of our proposed model with some existing pre-trained models

Table 6 displays comparisons between our proposed model and some of the previous published findings by other authors. Our model is the most reliable. One of the drawbacks of the current models was their poor accuracy. In comparison, our model fared better than similar models. The accuracy of a few pre-trained models that were evaluated using the same dataset as our study is displayed in Table 4. We evaluated our model's accuracy, ROC curve, Precision, and Recall against that of Inception(googlenet), ResNet50, Mobilenet50, and VGG16. Table 4 shows us that out of all the pre-trained models, ours has the best accuracy, ROC curve, Precision, and Recall. When our model's accuracy is compared to other models written by other authors and pre-trained models already in use, it performs the best and achieves the maximum accuracy. As such, we can state with confidence that our model performs exceptionally well in the classification of the leaf disease. Web Application Interface Output: A leaf disease web application is a software tool accessible through a web browser that enables users to identify of classification and diseases affecting plant leaves. These applications typically utilize image processing to analyze images of leaves uploaded by users and provide information on.potential diseases.

User Interface (UI): Image Upload: Users can upload images of leaves through various methods such as file uploads, drag-and-drop functionality, integration for real-time capture.

Results Display: The application provides the user-friendly manner, often highlighting affected areas on the leaf image and offering classification of the identified disease.

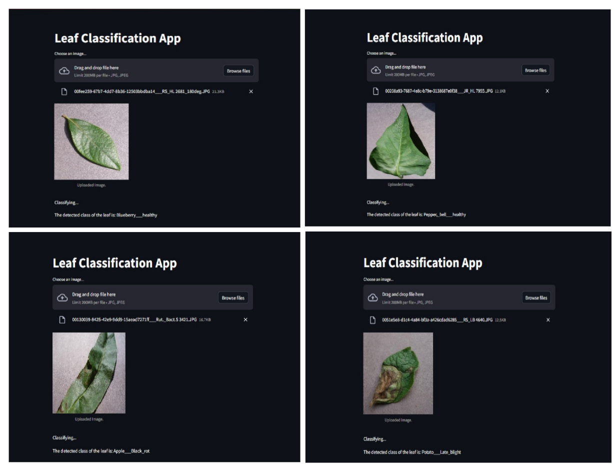

Fig. 12: Web application interface output.

F. Summary

This section presents a thorough analysis of the performance of the proposed model, which was evaluated using several metrics to determine how well it performed in classification tasks. Accuracy Accuracy,ROC Curve, Precision, Recall, were among the metrics used to provide a comprehensive picture of the model's predictive power. The section also included a comparison of the suggested model with other models that are currently in use in the same field. The comparison analysis's findings confirmed the suggested model's competitive advantage in classification tasks by highlighting the higher accuracy it attained. The model also performed better than established benchmarks, obtaining the best accuracy, when benchmarked against pre-trained models. These curves provided important insights into the model's training process and its capacity to generalize to new, unknown data by showing its learning dynamics over epochs. Strong performance in precision, recall, and ROC curves, along with high accuracy, highlight the suggested model's resilience to classification problems in the given domain. These results place the model in a competitive and promising position within the larger field of machine learning applications

V. Conclusion

In this study, we employed a dataset sourced from Kaggle to rigorously test and evaluate a leaf disease classification algorithm. The choice of dataset from Kaggle ensures a diverse and well-structured collection of images, fostering a robust assessment of the performance of the algorithm. To train and test our system precisely, we leveraged Resnet50 techniques, sophisticated and state-of-the-art convolutional neural network architecture. This choice is rooted in the effectiveness of Resnet50 in handling complex image classification tasks, making it particularly suitable for image analysis. It is worth noting that through our experimentation, the leaf disease classification algorithm exhibited superior performance when utilizing the Resnet50 technique. The inherent characteristics of Resnet50, such as its ability to capture intricate features and patterns, contribute to enhanced accuracy and reliability in disease categorization. A notable contribution of our work lies in the application of the Resnet50 method to categorize curl viruses, a novel aspect that, to the best of our knowledge, has not been explored in previous architectures. This shows the adaptability and versatility of the Resnet50 technique in addressing a diverse range of diseases beyond the primary focus on classification and leaf disease. To provide a comprehensive assessment, we conducted a comparative analysis of the performance of the proposed architecture with other existing architectures. This comparison involves evaluating key metrics, such as accuracy, precision, recall, and ROC score. The goal was to highlight the strengths and potential advancements introduced by our proposed architecture in the realm of image classification. In conclusion, the utilization of the Kaggle dataset, coupled with the Resnet50 technique, forms the backbone of our study in leaf disease classification. The exploration of curl virus categorization using Resnet50 adds a novel dimension to this research. The subsequent performance evaluation compared to other architectures contributes to a nuanced understanding of the effectiveness and uniqueness of the proposed system in the broader landscape of image analysis.

Future works and Direction : The progress made in this classification of leaf disease endeavor opens up new avenues for investigation and growth. Although the accuracy of the present model is impressive, there are still some areas that could be further researched and improved.We want our leaf disease classification models to run smoothly, even on simple devices such as smartphones or sensors used in the field. To do this, we look into ways to make the models smaller and faster without losing accuracy. Techniques such as reducing the size of the model, cutting out unnecessary parts, and learning from smaller models could help achieve this goal. This would make it easier to use our models in real-world farming situations, where resources are limited Below are written activities We will have future works.In addition to improving classification accuracy, our future work will focus on integrating real-time monitoring and automatic warning systems. This helps provide immediate feedback to farmers when a disease is detected, allowing timely intervention and disease management. This integration will help farmers increase crop production and reduce crop loss.

Finally, successful deployment in real-world scenarios requires close collaboration with domain experts such as agronomists, plant pathologists, and farmers. We plan to partner with these stakeholders to gather their feedback, validate our models, and ensure that our system performs well in terms of the specific requirements for different crops and regions.

References

[1] Ahmad, Aanis and Saraswat, Dharmendra and El Gamal, Aly, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, p. 100083, 2023.

[2] Singh, Vijai and Sharma, Namita and Singh, Shikha, "A review of imaging techniques for plant disease detection," *Artificial Intelligence in Agriculture*, vol. 4, pp. 229–242, 2020.

[3] Singh, Davinder and Jain, Naman and Jain, Pranjali and Kayal, Pratik and Kumawat, Sudhakar and Batra, Nipun, "PlantDoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pp. 249–253, 2020.

[4] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, p. 707, 2021.

[5] M. V. Applalanaidu and G. Kumaravelan, "A review of machine learning approaches in plant leaf disease detection and classification," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 716–724, IEEE, 2021.

[6] Panchal, Adesh V and Patel, Subhash Chandra and Bagyalakshmi, K and Kumar, Pankaj and Khan, Ihtiram Raza and Soni, Mukesh, "Image-based plant diseases detection using deep learning," *Materials Today: Proceedings*, vol. 80, pp. 3500–3506, 2023.

[7] Swaminathan, Akshay and Varun, C and Kalaivani, S and others, "Multiple plant leaf disease classification using densenet-121 architecture," *Int. J. Electr. Eng. Technol*, vol. 12, pp. 38–57, 2021.

[8] Atila, Ümit and Uçar, Murat and Akyol, Kemal and Uçar, Emine, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021.

[9] Sachdeva, Guneet and Singh, Preeti and Kaur, Pardeep, "Plant leaf disease classification using deep Convolutional neural network with Bayesian learning," *Materials Today: Proceedings*, vol. 45, pp. 5584–5590, 2021.

[10] Patil, Shanta and Vinay, SR and Kulkarni, Kartik and Mulla, Mehboobi and Naveen, Amruta and Beelagi, Sahana, "Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques,"

[11] Al-Wesabi, Fahd N and Albraikan, Amani Abdulrahman and Hilal, Anwer Mustafa and Eltahir, Majdy M and Hamza, Manar Ahmed and Zamani, Abu Sarwar, "Artificial Intelligence Enabled Apple Leaf Disease Classification for Precision Agriculture.," *Computers, Materials & Continua*, vol. 70, no. 3, 2022.

[12] j. Rashid, Javed and Khan, Imran and Ali, Ghulam and Alturise, Fahad and Alkhalifah, Tamim and others, "Real-Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique.," vol. 74, no. 1, 2023.

[13] Upadhyay, Santosh Kumar and Kumar, Avadhesh, "A novel approach for rice plant diseases classification with deep convolutional neural network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185–199, 2022.

[14] Diana Andrushia, A and Mary Neebha, T and Trephena Patricia, A and Umadevi, S and Anand, N and Varshney, Atul, "Image-based disease classification in grape leaves using convolutional capsule network," *Soft Computing*, vol. 27, no. 3, pp. 1457–1470, 2023.