

Custom CNN Model for Mango Leaf Disease Detection

Shiba Prasad Debnath¹, Pallab Chanda², Satyabrata Bhowmik³, Muskan Sutradhar⁴, Mission Debbarma⁵, Rupanjal Debbarma⁶

Department of Electronics and Communication Engineering Tripura Institute of Technology

DOI: <https://doi.org/10.51244/IJRSI.2026.1304000244>

Received: 22 April 2026; Accepted: 28 April 2026; Published: 19 May 2026

ABSTRACT

Plant health disorders can affect the productivity of crops adversely, so it's important to find them early in farming. This research develops a deep learning-based system for the identification of mango leaf diseases utilizing image data. We built and trained a custom Convolutional Neural Network (CNN) from scratch on the Mango Leaf BD dataset, which has eight types of healthy and diseased leaves. For making the images more generalized, they have gone under the procedure of resizing and normalizing before the data augmentation techniques are used. Standard evaluation criteria like accuracy, precision, recall, and F1-score are utilized to test the model, and it does well on the test dataset. Also, a desktop-based graphical user interface (GUI) is made with Python and Tkinter, which makes it easy to make predictions for one image or a group of images. The system works completely offline, so it can be used in places with few resources. It can also be expanded for use in real-world farming situations.

Indexterms- CNN(Convolutional Neural Network), Deep Learning, Image Classification, Mango Leaf Disease Detection, MangoLeafBD Dataset, Plant Disease Detection, Custom CNN, TensorFlow, RGB Image Processing, Graphical User Interface (GUI)

INTRODUCTION

In the economy of a country agriculture is very important because the people need to have enough food for survival, as the people in a country mainly depend on farming. Among various crops, mangoes are grown on a large scale and are very important to the economy. Different kinds of leaf diseases have always affected mango plants Often. If these diseases aren't found at an early stage, they can reduce the yield of the fruit. Traditionally, people predicted the diseases by looking at the leaves of plants by naked eye. This method heavily relies on human knowledge and can consume long time, especially when working with big farms. In lots of cases, early signs are either missed or misunderstood, which makes it harder to get the right treatment and leads to more crop loss. Recent improvements in deep learning and technology, primarily in image-based classification, has made it possible to automate the process of finding diseases easily. Convolutional Neural Networks (CNNs) have demonstrated strong performance in identifying patterns from images and are widely used for various computer vision tasks. These models can process important visual features such as color changes, spots, and texture variations that are commonly associated with plant diseases.

This project we are working on is solely dedicated to developing a CNN based model in which the mango leaf diseases can be detected using image data without the help of onsite manual work. Unlike many existing approaches that rely on conventional models, the proposed model is built and trained entirely from scratch. This allows better control over the architecture and provides more transparent understanding of how the model learns from the dataset.

The model is trained using a publicly accessible dataset and evaluated using standard performance metrics. In addition, a simple desktop based graphical interface is developed to make the system easily accessible for testing and demonstration purposes. The main goal is to create a reliable and practical solution for detection of plant diseases that can be further extended for real-world applications. Many recent works have demonstrated that deep learning, especially CNN-based models, can detect plant diseases more effectively, as they are able to learn useful patterns directly from image input . [1], [2].

Background and Motivation

Mangoes can be grown very easily and are cultivated as staple fruit in some countries and that's why it is important to the country's agricultural economy. However, mango plants are highly prone to various leaf diseases such as anthracnose, bacterial canker, and powdery mildew. Leaf disease does not only degrade the appearance of leaves but also reduce the overall productivity and quality of the crop. Early detection of such diseases is therefore necessary to minimize losses and ensure better yield and productivity.

Farmers still rely on manual inspection which gives slow outcomes and also always need the presence of an experienced individual for visual judgement.

In large-scale farms, it becomes difficult to monitor all plants on the regular basis, and early symptoms might go unnoticed. Ultimately, more plants may be vulnerable to the disease before proper action is taken. With increasing availability of image data and advancements in deep learning, the interest in using automated methods for plant disease detection has also increased. Convolutional Neural Networks (CNNs) have proved to be effective in analysing images and identifying patterns that may not be easily visible to the normal individual.

This makes them suitable for detecting visible changes in leaf morphology caused by the diseases. The motto behind this work is to develop a simple, efficient and effective system that can detect the mango leaf diseases using image data on its own. By using a custom CNN model trained from scratch, this study aims to show that accurate results can be achieved without depending on complex pre-trained models. Moreover, the system is also supported by a user-friendly interface, making it easier to detect and apply in practical scenarios

Problem Statement

Mango leaf diseases can significantly affect crop yield and quality if not identified at an early stage. In current practice, disease detection is mostly done through direct human review, which is time-consuming and depends on the past experience of the observer. This makes it tougher to ensure consistent and precise detection, especially in large scale farming systems.

One more limitation is the unavailability of simple and accessible tools that can directly identify different types of diseases from leaf images. While some deep learning-based approaches exist, many of them still depend on complex models that are already pre trained or require high computational resources, which may not be practical for all consumers.

Furthermore, several other factors in image conditions such as lighting, orientation angle, and background can affect the efficiency and performance of detection systems. This leads to the need for a more reliable model that can handle such variations and still provide reliable predictions of diseases in mango leaves.

Also, as the main challenge of this work is the way that disease affects the tree leaves and the answer to it is to create a customized CNN-model which will work to classify the disease Accurately and precisely and hence we can work on the plant for the further curing process. The system should be able to work with standard RGB images and should be easily portable and deployable in a simple desktop environment without requiring internet connectivity or specific equipment.

Objectives of the Study

The aim of this study is to develop a system, that is productive and reliable for detecting mango leaf diseases using deep learning techniques. The main aim is to create a model which can easily classify leaf images into various categories of diseases that it can be suffering from, based on the images provided. The Convolutional neural network (CNN) is a customized, designed & trained with the help of a dataset that can be accessed by public easily. Various techniques like image refinement and augmentation processes are present to improve the performance of the model such that the variations in the input images can be processed in a better way. The model is tested using standard metrics like accuracy, precision, recall and F1 score to ensure reliability.

In addition to model development, the other objective is to implement a complete prediction pipeline along with a simple platform specific desktop interface. This allows users to adequately use the model using both single image and batch prediction modes. The system is designed in such a manner that it can operate in absence of internet, that makes it practical to be used in the real world.

Contributions of this Paper

The deep learning based approach for mango leaf detection using a custom CNN model is presented by this paper. There are many methods that already exist which rely on pre trained architectures, but the model demonstrated in this work is entirely made from scratch, that provides more efficient control over design and training process.

An additional contribution lies in the development of a structured CNN architecture with multiple convolutional layers, incorporating regularization methods such as batch normalization and dropout to improve generalization and classification performance. These choices help the model perform better on the dataset.

The work also includes the development of a full system pipeline which is designed to automatize complex processive from ingestion to final delivery, covering data preparation, model training, evaluation, and prediction. Also, a graphical interface that is user friendly is integrated to demonstrate the uses of system in real life scenarios. The study shows that a tailored CNN model can efficiently detect plant diseases utilizing data that is publicly accessible.

Paper Organization

The rest of the paper is put together in a clear way. Similarly, the work, the dataset, and the methodologies used in this study are shown in the next sections. After that, the design of the CNN model and the training setup are explained. The analysis and discussion for this project and experimental observations are shown in the later sessions. The paper concludes with important findings and ideas for future work and scope.

Related Work

Image based plant detection has been popular these recent years and significant amount of research is still going on for improvement. The methods have changed from traditional image processing techniques to cutting-edge deep learning models, each with its own level of accuracy and difficulty. Earlier studies have explored different methods for plant disease detection, but recent work shows that deep learning approaches generally give better and more consistent results when working with image data [1], [2].

For years different type of research has also compared a significant amount of deep learning models and found that CNN- based models are performing better for plant disease classification tasks [3],[4]. In some cases, transfer learning has been used to improve performance, especially when the dataset size is limited [5].

Data augmentation techniques are also commonly used in image-based models to improve generalization and reduce overfitting during training [6].

Traditional Image Processing Methods

Traditional image processing methods were used in the early times for finding plant disease mostly. These traditional methods used steps like color segmentation, edge detection and texture analysis to find infected areas on the leaves. Features were manually taken based on things like shape and color changes. These methods were helpful for simple classification tasks, but they had some limitations. Their performance depended a lot on the lighting and background, and they often needed to know a lot about the domain to come up with good ways to extract features. Because of this, in complicated or real-life situations the accuracy was not very much or often the outcome was wrong.

Machine Learning Based Approaches

There were always some limitations of manual feature extraction. that is why machine learning techniques were introduced. Using these approaches, features retrieved from images were used to train classifiers such as k-nearest Neighbours(k-NN), Support Vector Machines (SVM), Decision Trees.

These methods improved classification performance compared to traditional techniques, as they were able to learn patterns from data rather than depending entirely on manually defined instructions. However, they still depended on the quality of the extracted features. Designing effective features remained a challenging task, and the overall performance was often limited by this dependency.

CNN Based Plant Disease Detection

In light of recent advancement in deep learning, Convolutional Neural Networks (CNNs) have become the most widely used approach for plant disease detection. CNNs are capable of automatically extracting features directly from images, eliminating the need for manual feature engineering.

Many studies have applied CNN models to datasets such as Plant Village and have achieved high accuracy in disease classification tasks. Some works also use transfer learning with pre-trained models like VGG, ResNet, and Mobile Net to further improve efficiency, even when the dataset is limited.

Compared to conventional and machine learning methods, better accuracy and robustness is provided by CNN-based methods as complex patterns and variations can be captured more conveniently. This makes them more reliable for hands on scenarios

Existing Work on Mango Disease Detection

There is limited research on mango leaf diseases as compared to other plant disease studies. Some of the existing works have applied deep learning and machine learning for the identification of the diseases such as powdery mildew and anthracnose in mango leaves.

Datasets or pre-trained models that are smaller have been utilized in such cases for categorization. While these techniques provide promising results, there is still a requirement of models that are trained particularly for mango leaf datasets that can perform efficiently without relying on transfer learning.

Research Gap

From the existing literature, it can be observed that although deep learning models have significantly improved plant disease detection, there are still certain voids. Many approaches depend on pre-trained models, which may not always be suitable for specific datasets or lightweight applications.

Furthermore, much work has not been done to develop a customized CNN model which is trained from scratch to detect mango leaves disease using well-organized dataset. For practical scenarios there is a need for simplified & portable system that can operate without internet access.

This study aspires to address these voids by developing a custom CNN-based model proficient from a publicly available mango leaf dataset, along with a comprehended system for predictive analysis and user interaction.

Dataset Description

This specific section describes the dataset that is used to train and test the model. To help the model learn and understand particular diseases better, a balanced and structured dataset is important for reliable results during classification.

MangoLeafBD Dataset Overview

The study focuses on mango leaf disease classification using an online publicly available dataset, the MangoLeafBD dataset. The dataset contains various disease images, where each image is classified into healthy and diseased classes. All the images of the mango leaf diseases were taken under controlled conditions [8]. The dataset contains multiple disease types, including healthy leaf samples suitable for multi-class classification. This dataset provides a good foundation for training a deep learning model as we can consider the images are well-balanced and clean.

Disease Class Description

Anthracnose is a common fungal disease primarily caused by the fungus *Colletotrichum gloeosporioides*. The first signs of disease are dark, uneven lesions or necrotic spots on top of the leaf. The spots on the leaf are typically surrounded by a yellow halo and get bigger as the infection progresses. When the air is hot and humid, the fungal spores spread quickly through rain and wind splash. This causes blossom blight and affects the fruit, causing it to rot after harvesting and a considerable amount of defoliation. Identifying anthracnose symptoms early on in leaves is important so that fungicide can be applied on time to save the crops. This class includes a total of 326 images in the dataset.

Bacterial canker is considered a severe bacterial disease that is primarily caused by *Xanthomonas campestris*. Diseased leaves first show signs like tiny, water-soaked, angular spots on the leaf. The affected spots grow larger rapidly and turn from dark brown to black, along with a clear yellow margin or halo around them. Early leaf drop and significant leaf blight can occur under persistent wet conditions, causing the lesions to merge. In plantations the disease is primarily hard to control, as it spreads quickly through rain blown by the wind and through infected plant materials or cutting tools. Severe outbreaks can cause loss of yield and tree decline. There are a total of 343 images in this dataset that represent this class.

Cutting weevil is caused by the adult mango leaf-cutting weevil, *Deporaus marginatus*. A female weevil lays her eggs by cutting across the mid-rib of young, soft mango leaves in a particular way. This makes the end leaf wilt, fold down, and eventually dry out completely. The leaves that have been affected have a unique look, with a cut or notch close to the central vein. This makes this class noticeable apart from fungal or bacterial diseases. During the flushing season a lot of new leaves can fall off trees if they are heavily infested. This makes the tree weaker and ineffective at photosynthesizing. There are 307 pictures of this class in the dataset

Die Back is a fungal disease mostly caused by *Lasiodiplodia theobromae* a common opportunistic pathogen. The disease move from top of the plant down to the bottom, starting with the tips of shoots turning dark and drying out. This process slowly moves down towards the main branches. Leaves that are affected turn die, brown and look like they have been burned. They usually stay attached to the shoot even after they die. The pathogen becomes more active when there is heat stress, drought, or mechanical injury. If it's not treated, it can cause whole branches to die and canopy to thin. There are 320 images of this class in the dataset.

Gall midge infection is caused by the mango gall midge, *Erosomyia mangiferae*; it is a small dipteran pest that specifically targets young mango foliage. The larvae excavate into the soft tissue of the leaf and consume from inside, which makes the plant grow bizarre, gall-like bumps or raised blister-like structures on the top of the leaf. Leaves that are infected generally show puckering, curling, and distortion, with the galls mainly showing up alongside the midrib and lateral veins. As the infection gets worse, the impacted tissue changes colour, turning either black or brown, which stops the normal photosynthesis process and slows down leaf growth. There are a total of 319 pictures of this class.

Powdery mildew is a fungal disease caused by *Oidium mangiferae*, a biotrophic pathogen that needs average temperatures and low humidity to grow. A distinctive white to greyish powdery fungal growth on upper and lower surfaces of leaves, along with on shoots and inflorescences,

Is a clear sign of the disease. Young leaves are particularly weak, and substantial infections can cause the leaves to bend, change shape, and fall off too early. Powdery mildew can cause plenty of flowers to die and a steep

decline in fruit set when it affects flowers and panicles. This makes it one of the most crucial diseases of mango in tropical and subtropical regions. This class has minimal images in the dataset (206), but it is still a crucial objective for automated disease detection.

Sooty Mould is caused by various saprophytic fungal species, mainly *Capnodium mangiferae*.

Which colonize the honeydew excreted by sap-sucking insects like mealybugs, aphids, and scale insects. One of the most distinct signs of the disease is that it makes the top side of mango leaves look like they are covered in dense, black, soot-like fungus. Sooty mold doesn't straight away infect plant tissue, but its dense coating blocks sunlight and makes photosynthesis inefficient. This causes the leaves to weaken, turn yellow too early and trees to grow more gradually at a slower pace over time. To stop it from happening repeatedly, it is crucial to manage the insect vectors well. There are 256 pictures of this class in the dataset.

Healthy leaf images are an important reference class. Which shows that the mango leaves don't show any concerning signs of disease, such as blight damage, or any other biological disorders. Healthy leaves are defined by deep green colour, distinctive lateral venation, a smooth upper layer, and an entire leaf margin; these are some of the characteristics which show a healthy and normal leaf shape. To recognize the difference between diseased and healthy leaves, it's important to add a healthy class in training classifiers files. Including a healthy class in classifiers makes it easier to identify diseases in binary and multi-class configurations. The healthy class has a total of 250 images that provide balanced and unbiased negative references [9].

Dataset Statistics

The MangoLeafBD dataset contains an identical number of images in each class, where each class is properly structured and balanced for disease classification. In total, the test set includes 4000 images, with each class containing 500 images across eight different categories. A balanced allocation of images is crucial, as it helps in lessening bias and assures that during training the model doesn't favor any particular class.

Having a structured and properly organized dataset provides stable assessment metrics, which include metrics like accuracy, precision, recall, and F1-score, thus producing more accurate and reliable results.

Train / Validation / Test Split

The dataset is equally divided into three segments: training, testing, and validation. The training and validation sets are important, as they are used to train the model and help track how well the model is understanding and prevent overfitting during the process. In the end, the testing set is used to check how the model performs on new, unseen data. The set is preserved separately and is only used in the end.

The train, test, and val subsets contain images that are specifically arranged into each class folder, which enables simplified loading utilizing standard deep learning frameworks. During training the model can be vulnerable to the test data; the divided subsets ensure that the model is evaluated fairly.

Sample Images

Sample images include both diseased and healthy leaves in the dataset, which demonstrates variations like discoloration, spots, and texture changes on leaves. During training for the model to acquire meaningful and significant patterns, the visual differences in leaves are important.

Even though the images are taken under restricted conditions, they still provide valuable features for classification. However, real-world practical images might contain additional variations; therefore, it should be considered in future work.



(a) Sooty Mould



(b) Gall Midge



(c) Healthy



(d) Anthracnose

PROPOSED METHODOLOGY

This section is about the steps which is followed to perform mango leaf disease detection. It has the system architecture, preprocessing steps, data augmentation, and the design of the CNN model. A program needs to be built where an image can be uploaded to be processed to produce a final classification result. The aim is to keep the system both efficient and reliable in maintaining the prediction task.

System Architecture Overview

The system has a process where an image is taken from the dataset and then it goes through preprocessing where it is resized and normalized.

Once finished, the image is forwarded to the CNN model, which extracts the required features and predicts the class. The output is then given as probability values, and the class which has the highest accuracy is taken as the final result. This prediction and the confidence level, is shown to the user. In folder mode, the same process is repeated for multiple images, and the results are recorded for the user.

Image Preprocessing Pipeline

Before input of images into the model, a preprocessing pipeline is used to ensure consistency. Images are loaded using OpenCV, which reads them in BGR format. These images are then converted to RGB images which must match the required input of the CNN.

All images size are changed to a predetermined dimension of 224×224 pixels. It is important because the model requires inputs of a specific size. After resizing, pixel values are normalized to the range of 0 to 1 which are divided by 255. This helps in stabilizing the training process.

Finally, the image is changed into a suitable array format and a batch dimension is added so that it can be forwarded to the model for prediction.

Data Augmentation Techniques

To increase the performance and generalization of the model, data augmentation techniques are used while training is being performed. These techniques create variations of all the training images by not changing their class labels.

Common augmentation methods used in this process are rotation, zooming, horizontal flipping, brightness adjustment, and shifting. These augmentation methods ensure the model can learn to handle variations in image orientation, scale, and lighting conditions.

And also to remember that augmentation methods is used only for training and the validation and test sets are not required. Which helps in evaluating the model on real images and gives a more clear idea of its performance.

Model Design Approach

The model prepared for this project is a custom Convolutional Neural Network. And it is built using various convolutional blocks that can learn various features from the input images, For example: simple patterns to more complex features.

Each and every block in this code has convolution layers along with batch normalization, activation functions, pooling and dropout to help prevent overfitting. And when the model runs, the amount of filters increases and it can capture more detailed features.

After the feature extraction is finished, a global average pooling layer converts the feature maps to compact form, which goes through connected layers to get the classification. The whole design has a goal to maintain a good balance between accuracy and efficiency, which makes the model more practical for real use.

Cnn Architecture

This section is about the steps of Convolutional Neural Network which is used to classify mango leaf diseases. The model is required to get good performance and to keep the overall structure clear and easy to understand.

Convolutional Block Structure

The concept of this section is CNN design in which we have to build a model that can understand disease patterns from leaf images and do not depend on any pre-trained models. Instead of using transfer learning, the model is developed, so that the structure and learning steps can be better controlled and understood.

The network is maintained in a way that it learns features one by one. The initial layers store simple details like edges and colour changes, and the next layers focus on more complex patterns such as textures and shapes related to diseases. To keep the training stable and prevent overfitting, techniques like batch normalization and dropout are also added. This repeated block structure can make the model to progressively learn hierarchical features and also maintain a balanced architecture.

Complete Layer-wise Architecture

The network has an input layer that takes images of size $224 \times 224 \times 3$, which is followed by five convolutional blocks with various number of filters, which starts from 32 and doubling in each subsequent block.

After the convolutional layers, a global average pooling layer comes into action to convert the feature maps to a one-dimensional vector. After that a dense layer is used to perform classification.

The final layer uses an activation function to produce probability values for all of the eight classes. The class with the highest accuracy is taken as the predicted label.

Model Parameters and Complexity

The CNN model is created to balance accuracy and computational efficiency. The various convolutional layers are used to increase the ability to learn complex features, and methods such as pooling and global average pooling help to reduce the number of parameters.

Batch normalization increases training stability, and dropout reduces overfitting, which allows the model to generalize more effectively. The network may be relatively deep, but it remains manageable for computational requirements and can be trained using standard GPU resources.

Overall, the architecture has good performance and complexity, making it best for image-based disease detection works.

Training Configuration

This section explains the necessary steps and the parameters required for the training of the CNN model. Better configuration is required to make sure that stable training and good performance are done.

Experimental Environment

The model is trained using a cloud-based notebook environment with GPU support. The implementation is done using the Python language and also TensorFlow and keras libraries are used. We can use the GPU for 30 hours a week which helps to reduce training time and allows efficient use of the dataset.

All experiments are done in a controlled environment to make sure the result is consistent. The trained model is then exported and used in a desktop system for prediction using a graphical interface(GUI).

Hyperparameter Settings

Many hyperparameters are required to maintain training efficiency and model performance. The image size is fixed at 224×224 pixels, and the whole batch size is set to 32. The code is trained for 50 epochs.

The Adam optimizer has a learning rate of 0.001, because it gives a stable and efficient convergence. The loss function used is categorical cross-entropy, and is better for multi-class classification problems.

These hyperparameters are chosen based on standard practices and modified to achieve good performance on the given dataset.

Callback Configuration

To increase the performance of the training process, various callback functions are used. Early stop is used to check the validation accuracy and stop training when no improvement is observed for a number of epochs, which helps in avoid the overfitting and the unnecessary use of GPU.

A model checkpoint is used to store the best-performing model during training. This make sure that the model used for evaluation has the most validation performance.

A learning rate modification is also applied during training. When the validation loss does not increase for a sometime then the learning rate is minimized. This make the model to do smaller updates to its weights and improve performance in the final stages of training..

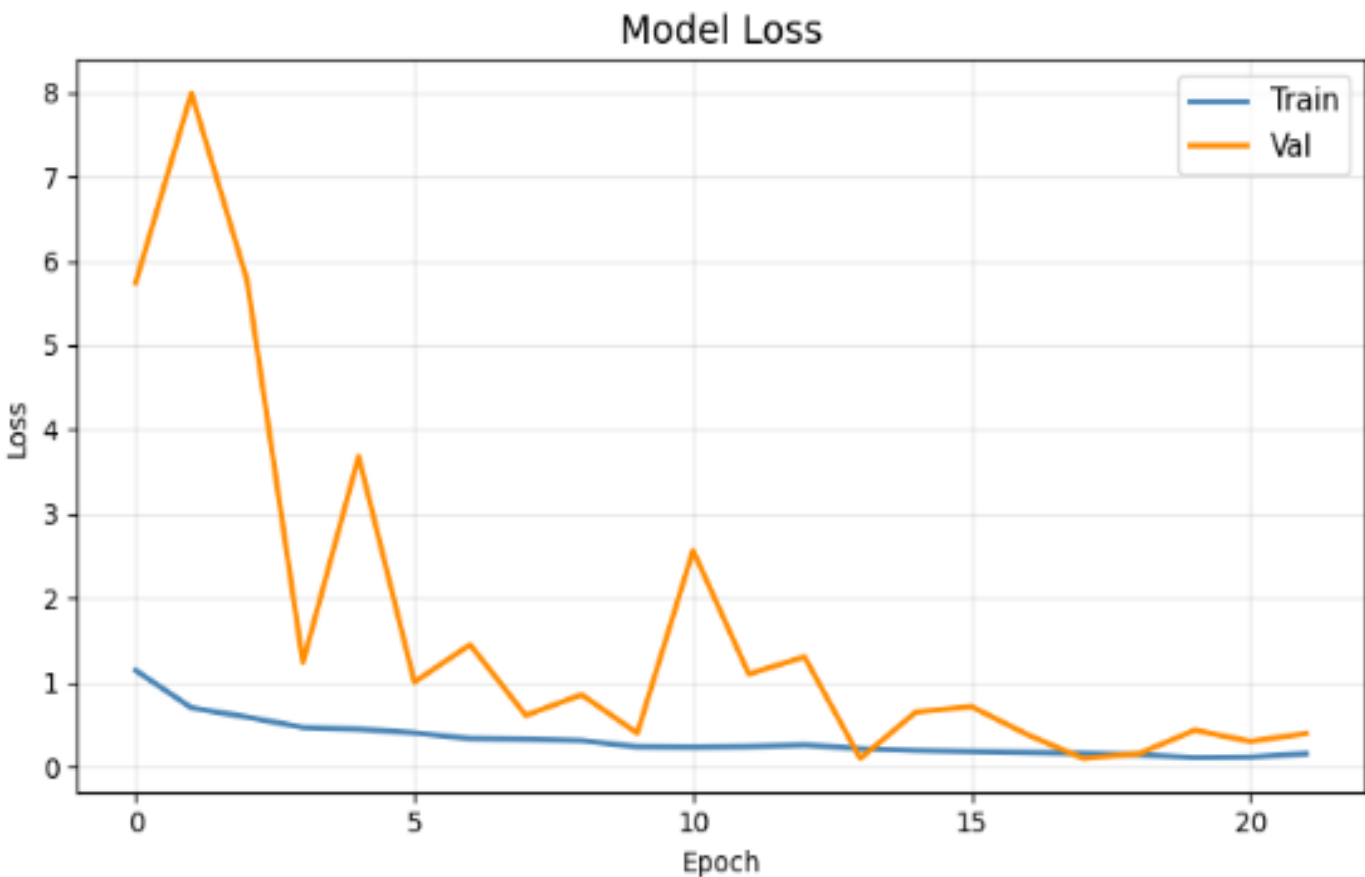
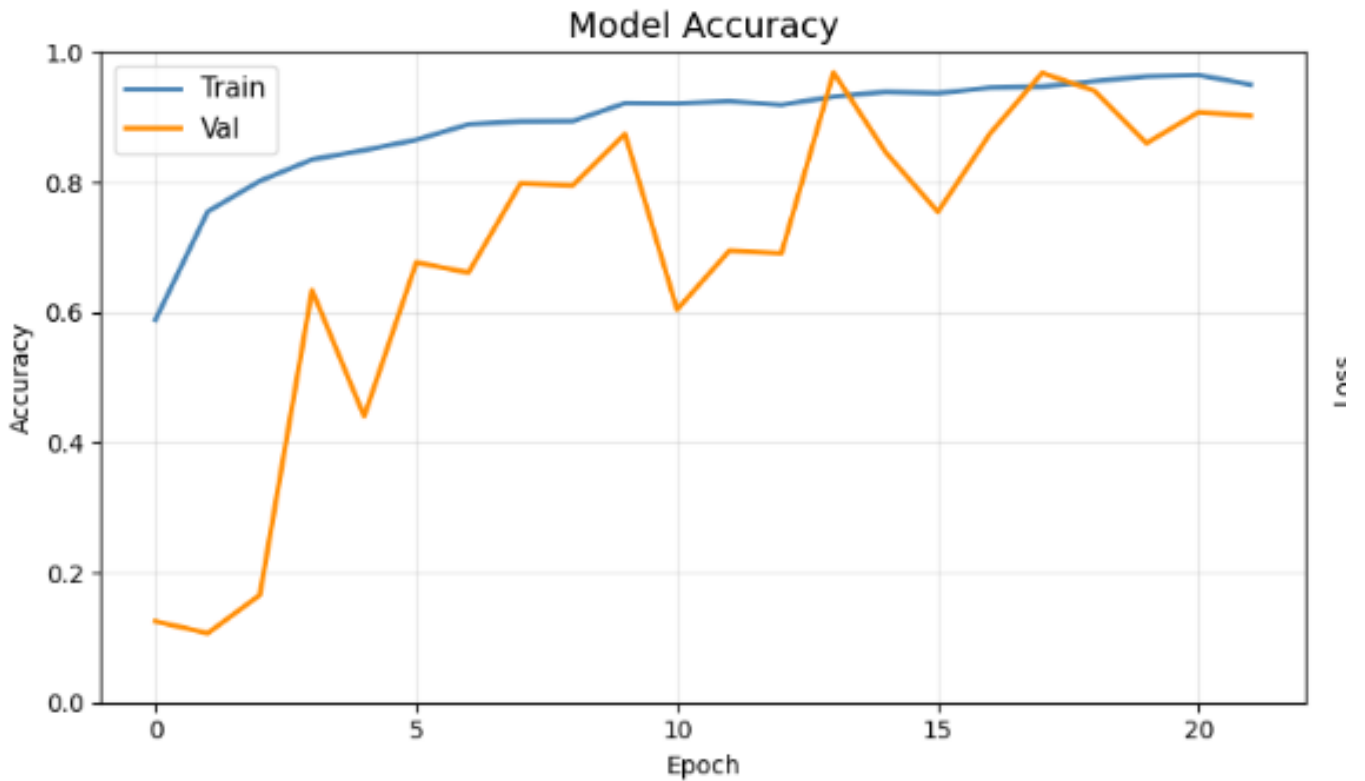
Training Procedure

The training starts when the dataset is loaded first and the required preprocessing and augmentation steps are applied. After that, the code is trained using the training data from the dataset, and while its performance is checked on the validation set to see how well it is learning..

During training, both accuracy and loss are tracked for the training and validation sets. These values are later used to plot graphs that show how the model is learning over time.

Once training is finished, the model is tested on a different dataset to check how well it performs on new, unseen images. After that, the final version of the model is saved so it can be used for making predictions..

CNN Training History



RESULTS AND EVALUATION

This section shows CNN model’s effectivity productivity on the dataset. The evaluation uses common metrics along with visual observations. It gives clear idea how accurately the model classifies different mango leaf diseases.

Training and Validation Accuracy

During the process, both validation and training accuracy shows improvement over epochs. The model understands significant patterns from the dataset. It also excels in accuracy. The validation accuracy adheres to a similar trend as the training accuracy, demonstrating that the model is not overfitting and is able to generalize well on unseen data. By the end of training, the model achieves very high accuracy on both training and validation sets.

Training and Validation Loss

The loss curves provide insight into the learning process. The training loss decreases persistently as the model understands from the data. The validation loss also shows a decreasing trend. And it also indicates stable training behaviour.

There is no significant contrast between training and validation loss. It suggests that the model is not memorizing the data but also learning general patterns. This confirms the accuracy of the selected system design and training system setup are Result-Oriented and successfully and practically efficient and productive.

Classification Report

The systematic grouping analysis report includes thorough extensive in-depth detailed performance metrics for each class. It also incorporates precision, recall, and F1-score. The represented system design structure framework accomplishes high values covering all these metrics. It showing strong systematic grouping performance.

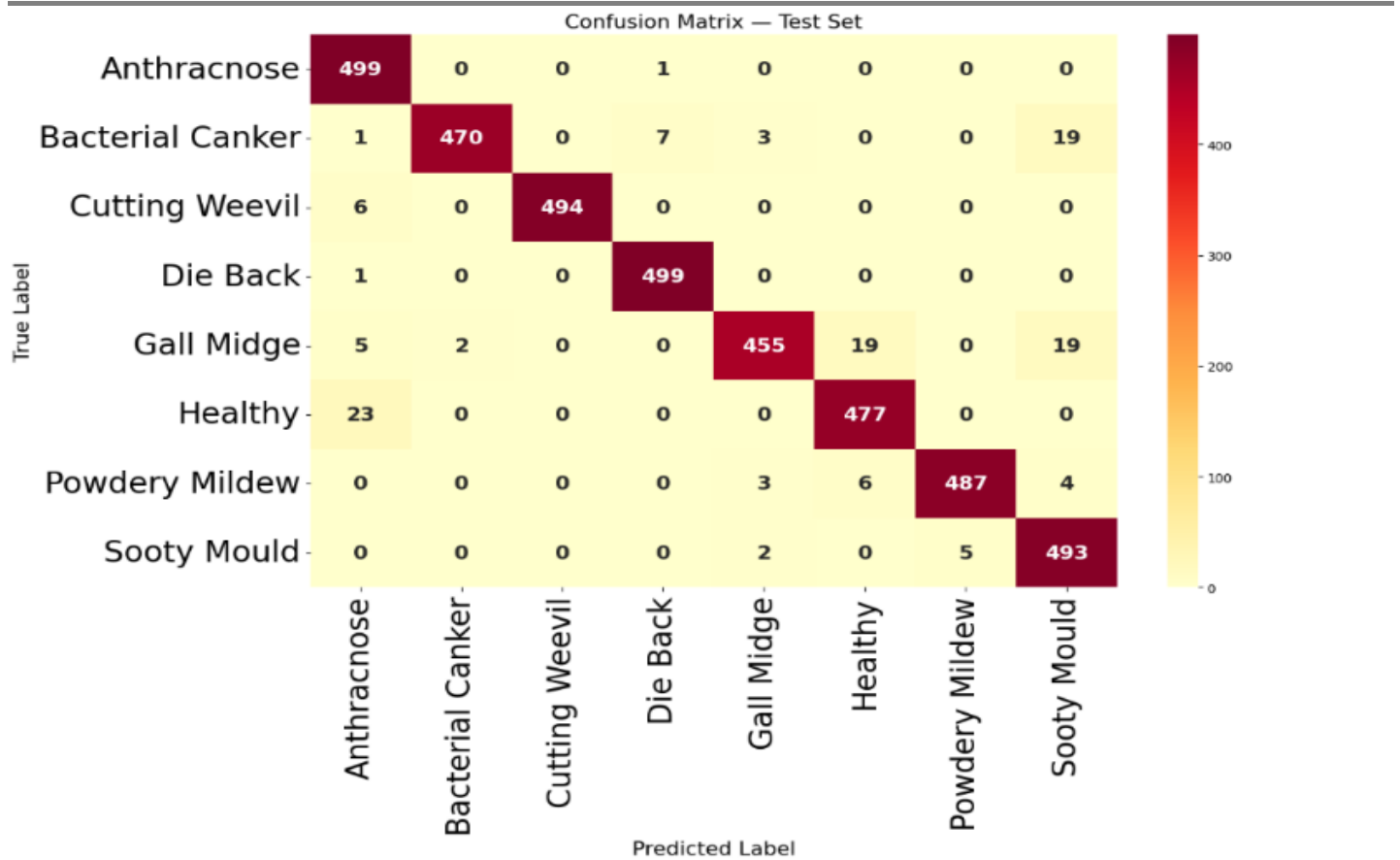
The results highlight that the model is able to clearly identify all classes. And it also has consistent output. This suggests that the dataset is well-organised. The model is effectively trained in classifying features of each category of disease.

Disease	precision	recall	f1-score	support
Anthraxnose	0.93	1.00	0.96	500
Bacterial Canker	1.00	0.94	0.97	500
Cutting Weevil	1.00	0.99	0.99	500
Die Back	0.98	1.00	0.99	500
Gall Midge	0.98	0.91	0.94	500
Healthy	0.95	0.95	0.95	500
Powdery Mildew	0.99	0.97	0.98	500
Sooty Mould	0.92	0.99	0.95	500

Confusion Matrix

The confusion matrix additionally confirms performance. The model presents the relationship between actual and predicted classes. The matrix is perfectly diagonal, shows that all samples are correctly classified.

This result demonstrates the effectiveness of this model in differentiating between the different disease classes. It also confirms the reliability of the classification report metrics.



Per-Class Performance Analysis

A detailed analysis of distinct classes demonstrates that the model performs equally well across all different types of categories. Each class achieves Persistent precision, recall, and F1-score values. It is demonstrating a stable performance.

This is mainly due to the balanced nature of the dataset.

Here each class contains an equal number of samples.

Therefore, the model does not show bias toward any particular class.

The results confirm that the Submitted CNN model is highly effective for mango leaf disease classification.

Graphical User Interface

To improve user experience, a desktop-based graphical user interface (GUI) is developed using Python and Tkinter.

GUI Design and Overview

The GUI is kept User-friendly so it is Intuitive to use. It has Distinct sections to load the model then choose images and then view the results along. The interface is divided into tabs. One for single image prediction . Another for processing multiple images. This layout makes it Transparent.

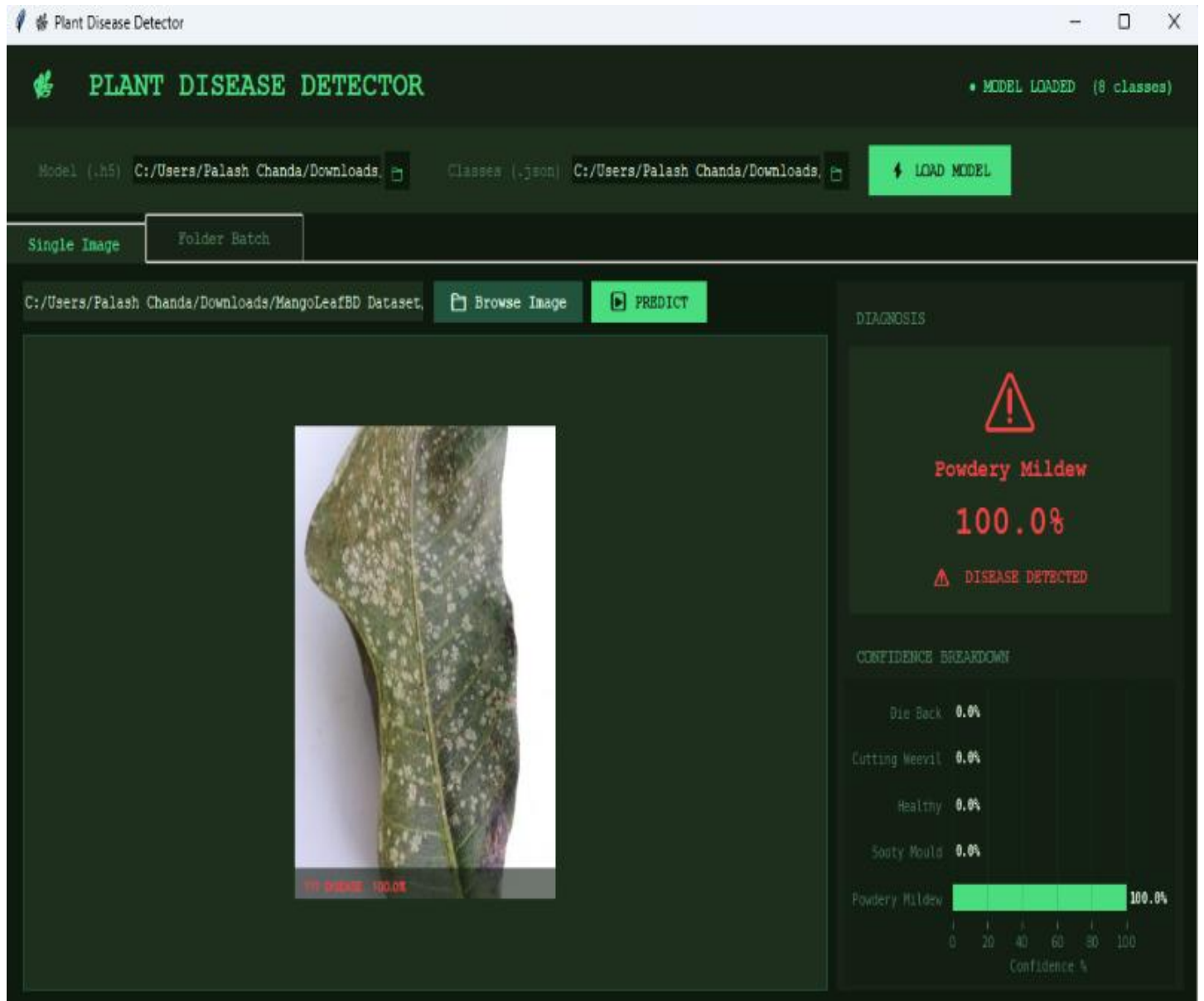
Model Loading Module

The system allows the user to load the trained model . The model is saved in a .h5 file .The class names are stored separately in a JSON file. This setup makes it easier to manage .And also update the system when needed. Users can choose these files manually through the interface. And also by the system. System can load them automatically. Once both files are loaded, the model is ready for the predictions.

Keeping the model and class labels separate also provides flexibility. When working with different datasets or updating the model, Keeping the model and class labels separate is helpful.

Single Image Prediction

The user selects a single leaf image, and then it is shown on the screen. After clicking the predict button, the image is Executed and Conveyed to the CNN model for Evaluation. The Mechanism then shows the predicted. It also shows the class along with a confidence value. It also Indicates a small chart of Likelihoods. As a result the user can get an Insight of how Reliable the model.

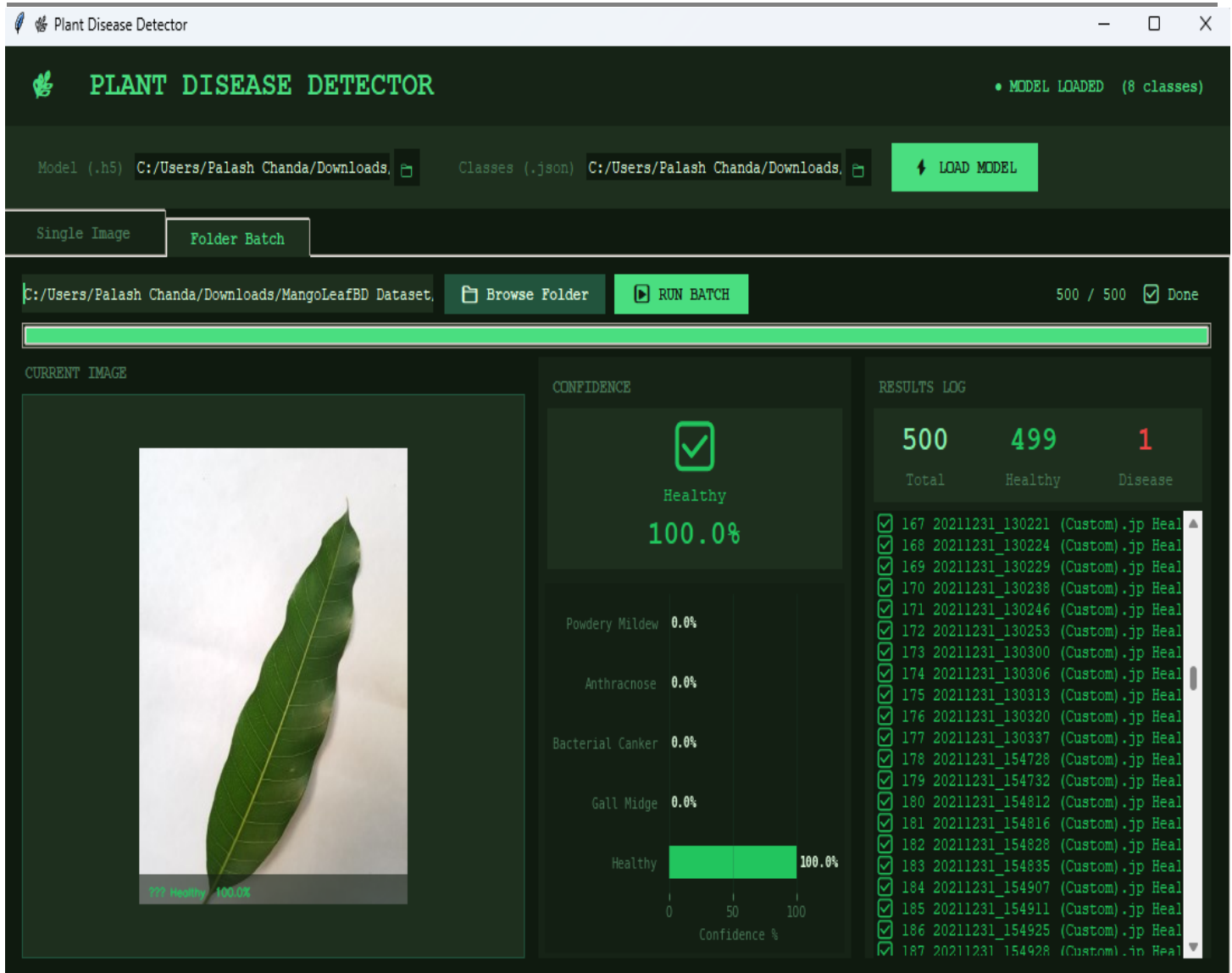


Folder Batch Prediction

The batch prediction, lets the user work with multiple images at once. It allows to simply select a folder.

The system Without manual input reads all valid image files inside. It processes them one by one. When the process runs, the interface displays Revisions like how many images have been Accomplished and also the Latest outcome.

It also Sustains a Record of ongoing activity of predictions for each image. It also makes it easier to review the outputs thereafter. This feature is Exceptionally useful. It is also helpful when testing the model on larger sets of images.



Results Display and Export

The Final outcomes are displayed In a clear and simple way. Each prediction displayed along with its confidence value. Basic Signs are also applied to assist and differentiate among healthy and diseased leaves. For batch processing, the outputs can be saved as a CSV file. It makes it convenient to check the results later. And also can be used for comparison and reporting. In general, the GUI makes the system easier to work with. It builds it easy, simple and user-friendly. It leads to providing a understandable and engaging and User-Interactive way to represent the model’s predictions..

DISCUSSION

This section provides an overall perception of the results obtained from the proposed system, along with comparisons, advantages, limitations, and possible future improvements. The results monitored in this work are in line with earlier studies, where deep learning models have shown strong performance for similar image-based classification tasks [1].

It has been monitored that in deep learning models similar trend has been followed where the plant disease classification tasks performance is strong [7].

Interpretation of Results

By the results it is confirmed that CNN model functions well in classifying mango leaf diseases. It achieves high accuracy with consistent exactness, recall, and F1-score beyond all classes. The training and validation trends

also remain stable, signifying that the model is learning useful patterns from the dataset rather than just memorizing it.

The model is able to discriminate between different disease classes with very few errors. This performance is mainly supported by the equitable dataset and the design of the CNN model.

Comparison with Related Works

Earlier methods relied on manual feature extraction and were sensitive to variations in image conditions.

Deep learning-based methods, especially those using CNNs, have shown improved accuracy in recent studies. However, many of these approaches depend on pre-trained models. In contrast, the model in this work is built entirely from scratch and still accomplish competitive performance.

Although pre-trained models may offer slight improvements in some cases, the proposed approach demonstrates that a custom CNN can accomplish strong results when trained on a well-structured dataset.

Advantages of Proposed Method

One of the main advantages of the proposed system is that it is built entirely using a custom CNN model without relying on pre-conditioned networks. This provides better control over the architecture and makes the system more versatile.

Another advantage is the clarity of the overall pipeline. The system uses standard RGB images and does not necessitate additional hardware or complex preprocessing techniques. The incorporation of a graphical user interface further improves practicality by allowing users to easily test the model.

The system is also designed to work in an offline environment, making it suitable for practical use in areas with limited internet access.

Limitations

Despite the strong performance some challenges will emerge eventually. The dataset used in this study comprises of images taken under relatively controlled circumstances. As a result, the model may not perform equally well on real-world images with complex backgrounds or shifting lighting conditions. Another limitation is that the model emphasized only on classification and does not provide meticulous information about the intensity or exact location of the disease on the leaf. In addition, the current work does not include comparisons with multiple models or sophisticated architectures, which could provide a deeper understanding of rendering differences.

Future Work

Future improvements can focus on testing the model with real-world images captured under different environmental conditions. This would help in estimating the robustness of the system.

The model can also be more sophisticated to augment more plant species and supplementary disease categories. Inclusion explainable AI techniques such as visualization of feature maps could equip better intuition into how the model makes predictions.

And also we can use it in mobile platform and use images taken from drone to improve real-time use which increases the practical use of the system.

The future of this project and research may also include the technique of comparing this model with different and new deep learning technologies and optimizing the implementation on mobile or web-based platforms.

CONCLUSION

A custom convolutional neural network (CNN) is proposed, a deep learning-based method for mango leaf disease detection. This model doesn't rely on any pre-trained architectures; it was trained and designed from scratch using the MangoLeafBD dataset. The result shows a high level of accuracy in classification; along with that, the model is able to capture significant disease-related patterns from leaf images. To improve the overall system performance, steps like data augmentation and preprocessing steps play a key role in enhancing the system. The MangoleafBD dataset is proportionate and balanced, which is important as it helps in maintaining consistent results across all the classes. When the model is trained on a proportionate and balanced image dataset, it can be used effectively for plant disease detection using deep learning methods [1].

A desktop-based graphical interface is developed to test both single-image and batch prediction alternatives in the system; it was developed along with the model. Although the trained model performs properly under the current setup, it needs to be tested for it to better understand more complex conditions of real-world images. Overall, for plant disease detection this system offers an easy-to-use and practical approach and can be used for wider agricultural applications.

Appendix

Before training, the images were normalized and resized to 224 x 224 pixels, along with TensorFlow and Keras, which were used to implement the model. After the model is trained, it is saved in .h5 format, whereas a separate JSON file is used to store class labels. The separation of class labels and the model allows for better flexibility during deployment and prediction.

Both OpenCV and NumPy are used by the prediction pipeline for image processing and handling data operations; Tkinter is used to develop the graphical interface, and libraries like Matplotlib and Pillow are used for proper visualization and image display.

ACKNOWLEDGMENT

For this work we would like to express our sincere thanks to our project guide and faculty members of our institution for their guidance and continuous support.

We would also like to thank our institution for a project-friendly environment and providing all the necessary resources for this work. Additionally, for training and evaluating the model, we acknowledge the use of datasets available online, which played an important role in this work.

REFERENCES

1. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, 2016.
2. K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, 2018.
3. E. C. Too et al., "A Comparative Study of Deep Learning Models for Plant Disease Identification," *Computers and Electronics in Agriculture*, 2019.
4. D. P. Hughes and M. Salathé, "An Open Access Repository of Images on Plant Health," *arXiv*, 2015.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
6. C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, 2019.
7. P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

8. Kaggle, "Mango Leaf Disease Dataset."
9. S. I. Ahmed, M. Ibrahim, Md. Nadim, Md. M. Rahman, M. M. Shejunti, T. Jabid, and Md. S. Ali, "MangoLeafBD: A Comprehensive Image Dataset to Classify Diseased and Healthy Mango Leaves," Data in Brief, vol. 47, p. 108941, Apr. 2023, doi: 10.1016/j.dib.2023.108941.