A Project Report on

Leaf Disease Detection and Recognition

Submitted in partial fulfillment for the award of

Bachelor of Technology

in

Computer Science & Engineering

By

M. Gnana Sireesha(Y19ACS499) **K. Subbarao (Y19ACS479)**

J. Ajay Raj(Y19ACS463) **Ch. Nitish(Y19ACS436)**

Under the guidance of

A. Gopinath, Assistant Professor

Department of Computer Science and Engineering

Bapatla Engineering College

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522 102, Andhra Pradesh, INDIA

2022-2023

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project report entitled **Leaf Disease Detection and Recognition** that is being submitted by **M. Gnana Sireesha (Y19ACS499), K. Subbarao(Y19ACS479), J. Ajay Raj (Y19ACS463), Ch. Nitish(Y19ACS436)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

P. Pardhasaradhi
Prof. & HOD

A. Gopinath
Assistant Professor

Declaration

We declare that this project work is composed by ourselves, and that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

M. Gnana Sireesha (Y19ACS499)

K. Subbarao(Y19ACS479)

J. Ajay Raj (Y19ACS463)

Ch. Nitish(Y19ACS436)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **A. Gopinath**, Assistant Professor, Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **P. Pardhasaradhi**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal, **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to the project coordination committee and our project coordinator **Dr. N. Sudhakar,** Prof., Dept. of CSE, **Dr. D Kishore Babu,** Asst. Prof., Dept. of CSE **Sri. M. Rajesh Babu,** Asst**.** Prof., Dept. of CSE for their helpful suggestions throughout the project work and in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

M. Gnana Sireesha (Y19ACS499)
K. Subbarao(Y19ACS479)
J. Ajay Raj(Y19ACS463)
Ch. Nitish(Y19ACS436)

Abstract

Agriculture plays a crucial role in the economy of most countries, and plant diseases can significantly affect crop yields and quality. Therefore, early detection of plant diseases is essential for effective disease management and minimizing economic losses. In recent years, deep learning algorithms, particularly convolutional neural networks (CNNs), have shown promising results in detecting various plant diseases. This paper proposes a novel approach for the detection of multiple leaf diseases using a CNN model.

The proposed approach consists of several stages, starting with the preprocessing of leaf images to extract relevant features. The preprocessed images are then used to train a CNN model using a dataset consisting of multiple leaf diseases. The dataset used in this study contains images of different leaves affected by different diseases.

My aim is to detect the type of leaf and disease that the leaf affected and also to recommend pesticides to cure leaf disease. In this we tend to area unit using image processing techniques to classify diseases & quickly diagnosis can be carried out as per disease.It includes several steps viz. image acquisition, image pre-processing, segmentation, features extraction and convolutional neural network-based classification. Aim of my project is to identify leaf type and detect the leaf diseases and provide recommendations of pesticides that helps to recover the leaf from the disease for various types of leaves.

In conclusion, the proposed approach for multiple leaf disease detection using CNN is a promising approach that can effectively detect multiple leaf diseases with overall accuracy of 90% and recommend pesticides.

Table Of Contents

List of Figures

List of Abrivations

1. Introduction

Multiple leaf disease detection and recommending pesticides using Convolutional Neural Networks (CNNs) is an advanced approach that automates the identification and classification of multiple diseases in plants and recommends suitable pesticides for the specific disease identified.

Plant diseases can have a significant impact on crop yield and quality, and identifying the specific disease and selecting an appropriate pesticide can be challenging and time-consuming. Traditional methods of detection and recommendation are often based on manual inspection and expert knowledge, which may not always be reliable. [1]

1.1 Background

In this approach, a CNN model is trained on a large dataset of images of healthy and diseased plant leaves, along with the corresponding pesticide that was used to treat the disease. The model learns to identify patterns and features in the images that are indicative of a particular disease and its corresponding pesticide.

Once trained, the model can be used to classify new images of plant leaves as healthy or diseased and recommend the appropriate pesticide based on the disease identified. This approach can help farmers quickly and accurately identify the disease affecting their plants and recommend suitable treatment, leading to timely intervention and reduced crop losses.

The success of the CNN model depends on the quality and quantity of the dataset used for training, as well as the accuracy and completeness of the pesticide information. Therefore, creating a large, diverse, and high-quality dataset of plant images, along with corresponding pesticide information, is crucial for the accuracy of the model.

Multiple leaf disease detection and recommending pesticides using CNNs has the potential to significantly improve the efficiency and effectiveness of crop management, helping farmers make more informed decisions and improving crop yields and quality.

1.2 Problem Statement

The problem statement for single leaf disease detection using Support Vector Machines (SVMs) is to develop a machine learning model that can accurately classify a given image of a single plant leaf as either healthy or diseased and, if diseased, identify the specific disease.

Traditional methods for identifying plant diseases involve visual inspection by experts, which can be time-consuming and expensive. Machine learning models, such as SVMs, offer a more efficient and cost-effective alternative to manual inspection. [2]

To develop the SVM model, a dataset of images of healthy and diseased plant leaves is required. The model is trained on this dataset to learn the patterns and features that distinguish healthy leaves from diseased leaves. Once trained, the SVM model can classify new images of plant leaves as healthy or diseased, with high accuracy.

Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. The studies of the plant diseases mean the studies of visually observable patterns seen on the plant. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertise in the plant diseases, and also require the excessive processing time. Hence, image processing and Deep learning techniques are used for the detection of plant diseases. [3]

1.3 Solution To Problem Statement

The proposed system makes use of one of the deep learning techniques called the Convolutional Neural Network (CNN). CNN are mainly used for classification problems. The proposed system uses convolution neural network to get better results. The system takes in an image of a leaf as an input through the device internal storage and goes through five level of classification described below:

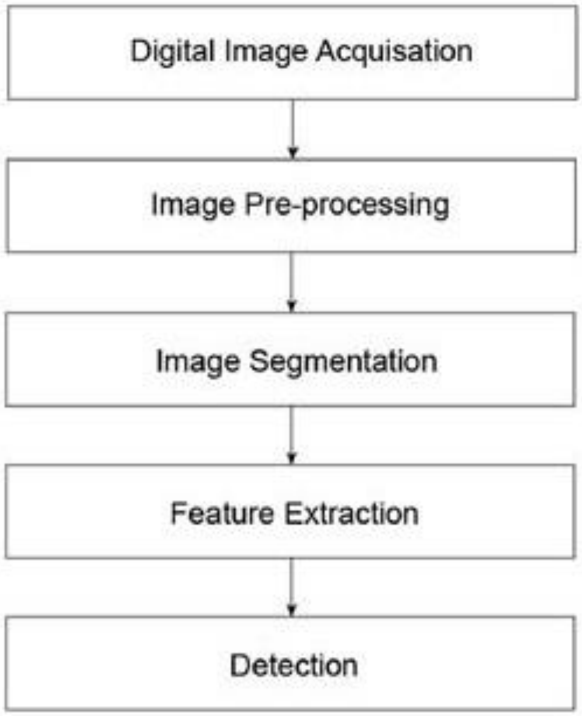


Figure 1.1Classification Of Image

Once the disease is been finally classified, its related description about the disease and pesticides is provided. The proposed system is capable of identifying the diseases of plants such as Tomato, Apple, Pepper, Potato, etc.

1.4 Objective

The objective of Plant disease detection using leaf images is to design an incremental model to detect the plant diseases ignoring external features like environment, noise and background. This focuses on identifying various diseases in plants and to ease the job of farmers or to educate the farmers about the disease detected. An approach of classification using Convolutional Neural Network that has very good working efficiency produces the accurate results. The system helps to improve the performance. Maintaining the project is easy and manageable. [1]

1. To enhance the given input image by Image acquisition and Image preprocessing.
2. Identify the affected part through texture analysis and Segmentation.
3. Classify the healthy and affected leaf part by feature extraction and classification.
4. Train the model by using testing data for accurate result.

2 Literature Survey

In recent years, object recognition and disease detection using deep and machine learning approaches has increased gradually. In agricultural research projects, many of these methods have been applied, like Artificial neural networks, K-means, Support Vector Machines (SVMs) etc. [10].

In 2019 [11], a convolutional neural network (CNN) had been used for tea leaf diseases where images were classified automatically. A dense scale-invariant feature transform was used to extracted features. For classifying diseases, support vector machine (SVM) and multi-layer perceptron (MLP) classifiers had been used.

In 2003, a method was proposed by Helly et al. [9] which could detect diseases from leaf images. Hue Saturation Intensity (HSI) transformation and image segmentation has been done with fuzzy C means. Neural network-based classification had been performed there for detection.

In recent years, object recognition and disease detection using deep and machine learning approaches has increased gradually. In agricultural research projects, many of these methods have been applied, like Artificial neural networks, K-means, Decision trees, Support Vector Machines (SVMs) etc. [10].

Prajwala TM et al. [17], they designed a system to detect and identify diseases in tomato leaves by implements, a slight difference in the convolutional neural network design called LeNet. The main objective of this work was to discover a solution to the difficulty of detecting tomato leaf disease using the easiest method while utilizing minimal computational resources to produce outcomes that were comparable to state-of-the-art methods. Neural network systems use an automated extraction of features to help differentiate the input image into several classes of illness. This system has obtained an overall accuracy of 94-95 %, showing even under unfavorable conditions the viability of the neural network approach

Image analysis was done by automated techniques that focus on the color features of objects and can be used to identify the edge of affected areas in order to identify the diseases. Author Amar Kumar [12], deals with leaf rot disease detection for betel vine using an image processing algorithm. Few vision-based methods had been used. Based on the color feature of the rotted leaf area diseases are detected. Segmentation, thresholding was part of the research. Though the paper proposed the fastest methods of image processing it only focused on one disease of betel vine leaf and other diseases had been ignored during research.

3 Existing System

The existing system follow support vector machine (SVM) algorithm and the Gaussian Mixture model for image segmentation to detect disease of leaf. This model only works for detecting multiple diseases of single leaf. Leaf image is given as input and is matched with pre trained data and detect type of disease. Leaves diseases caused by regular endangerment to bacteria which causes a huge yield loss globally. Machine learning, the latest breakthrough in computer vision, is encouraging for fine-grained disease classification, as the method uses SVM classifier and Gaussian mixture model for image segmentation. Disease detection and classifications are considered as the two hardest works to the recognition of disease. [4]

3.1 Support vector machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression analysis. SVM works by finding the best possible boundary that separates different classes in the dataset. In SVM, the data is represented as points in a multidimensional space, and the goal is to find a hyperplane that can separate the data into classes. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. [5]

SVM is particularly useful for handling high-dimensional data and is effective in cases where the number of features is much greater than the number of samples. SVM can also handle nonlinear data by using kernel functions to map the data to a higher-dimensional space where it can be linearly separated. SVM has several parameters that can be tuned to achieve better results, including the choice of kernel function, regularization parameter, and penalty parameter. SVM has been widely used in various fields, including image recognition, text classification, and bioinformatics.

3.1.1 Image pre- processing

The pre-processing stages are used to the given image to make it suitable for further processing. At this step, we convert the colour space from RGB to the LAB colour space. Because the model L*a*b* is a three-dimensional model it can only be properly designed in three-dimensional space. We convert to the LAB colour space to distinguish the affected areas by the leaf of the plant and segment them with the next step, which is segmentation the images.

3.1.2 Image segmentation

To distinguish the background and the affected area, the image segmentation is employed. For Image segmentation, two primary methods are used: cluster-based and colour based. The proposed system will give more trustworthy results using cluster subtraction by using K-mean clustering algorithm. The K-Means clustering method attempts to identify objects depended on a set of attributes into K number of classes. Classification shall be effected by reducing the sum of squares of distances between entities and the correlating group or class centroid. In our research, multiple values of the number of clusters have been examined. In Fig. 2, an example from our work illustrates the segmentation of images of tomato plant infected with the target-spot disease

3.1.3 image post-processing

After segment the image, we need a process called postprocessing to configure the data for the next step. In the proposed system and in this process, after image segmentation, the image is converted to the grayscale to extract the features from the image .

3.1.4 Feature extraction

Different features are extracted after the segmentation of the infected area to characterize the infected area. In this step, the important features are extracted from the image after converting it to the gray level. The most important features that were extracted in our work are the features that depend on the texture. Here we use several features that have been calculated from the image, and they are called statistical features such as mean, standard deviation, skewness, kurtosis, entropy, variance, and root mean square (RMS). Also, a Gray Level Co-occurrence Matrix (GLCM) method was used to extract some other important features. GLCM texture gets the connection between two pixels one after another, called the reference and the neighbour pixel. The features that were extracted from the GLCM method are energy, homogeneity, contrast, and correlation features. [6]

3.1.5 Classification

Classification is an important step. It is used to classify images depending on their features. In this step, we use the support vector machine (SVM) algorithm. SVM has been proposed only for two-class issues but also for decisionmaking. The multi-support vector machine (MSVM) because there are more than 2 classes. In the proposed system, there are 15 classes. An SVM builds a hyper-plane or collection of hyper-plans in a high- or limitless-dimensional space that can be used for classification, regression, or different functions. It is a supervised learning system with associated learning methods that investigate data. Then we divided the data into two are training and testing. To transfer training and testing samples to the high-dimensional feature space, kernel functions are implemented. The type of kernel function we were used is a Radial Basis Function (RBF). At the end of this step, the plant leaf diseases were detected and classified using the SVM method.

3.2 Gaussian Mixture Model

A Gaussian mixture model (GMM) is a probabilistic model that represents a probability distribution as a weighted sum of Gaussian probability distributions. It is a flexible statistical model that can capture complex patterns in data, including multi-modal distributions.

In a GMM, each component of the mixture represents a Gaussian distribution with its own mean and covariance matrix. The weights assigned to each component specify the importance or probability of that component in the overall mixture.

GMMs are often used in machine learning for applications such as clustering, density estimation, and dimensionality reduction. They can also be used for image processing and speech recognition.

The parameters of a GMM can be estimated using maximum likelihood estimation or Bayesian inference. Once the model has been trained, it can be used to generate new data points or to make predictions about new data.

3.3 Limitations Of Existing System

There are several limitations for single leaf disease detection using Support Vector Machines (SVMs):

1. Limited accuracy: SVM models can achieve high accuracy in classification tasks, but they may not always be the most accurate for detecting single leaf diseases. Other machine learning models, such as Convolutional Neural Networks (CNNs), may perform better in certain cases.
2. Dataset limitations: SVM models require large amounts of labeled data to be trained effectively. However, obtaining a large dataset of high-quality images

of single plant leaves with various diseases can be challenging and time-consuming.

3. Sensitivity to hyperparameters: SVM models are sensitive to their hyperparameters, which are parameters that are not learned from the data but must be set before training the model. Selecting the appropriate hyperparameters for the SVM model can be difficult and requires expert knowledge.
4. Limited interpretability: SVM models can be difficult to interpret, making it challenging to understand how the model is making its predictions. This can be problematic when trying to identify the specific features or patterns that are indicative of a particular disease.
5. Lack of robustness: SVM models can be sensitive to changes in lighting conditions, image orientation, and image quality, making the model less robust and prone to errors when working with real-world images.

Overall, while SVMs can be useful for single leaf disease detection, they have several limitations that need to be considered when choosing an appropriate machine learning approach.

4 Proposed System

The main purpose of proposed system is to detect the diseases of plant leaves by using feature extraction methods where features such as shape, color, and texture are taken into consideration. Convolutional neural network (CNN), a machine learning technique is used in classifying the plant leaves into healthy or diseased and if it is a diseased plant leaf, CNN will give the name of that particular disease. Suggesting remedies for particular disease is made which will help in growing healthy plants and improve the productivity. [7]

First the images of various leaves are acquired using high resolution camera so as to get the better results & efficiency. Then image processing techniques are applied to these images to extract useful features which will be required for further analysis.

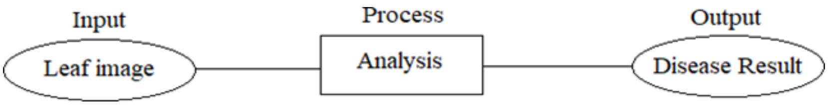


Figure 4.1 Flow Chart Of Proposed System

A proposed system for multiple leaf disease detection and pesticide recommendation using CNN (Convolutional Neural Network) could consist of the following components:

4.1 Dataset Collection

A large dataset of high-quality images of various leaf diseases should be collected. These images should include leaves infected with multiple diseases at the same time. A dataset of pesticides and their corresponding information should also be collected.

4.2 Pre-Processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Preprocessing data is a common first step in the deep learning workflow to prepare raw data in a format that the network can accept. For example, you can resize image input to match the size of an image input layer. You can also preprocess data to enhance desired features or reduce artifacts that can bias the network. [7]

4.3 Image Processing

Image processing is any form of processing for which the input is an image or a series of images or videos, such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to the image. It also means "Analyzing and manipulating images with a computer".

Image processing is performed by these three steps:

1. First, import images with a device like a scanner or a camera or directly through digital processing.
2. Second, manipulate or analyze the images in some way. This step can include image improvement and data summary, or the images are analyzed to find rules that aren't seen by the human eyes. For example, meteorologists use this processing to analyze satellite photographs.
3. Last, output the result of image processing. The result might be the image changed by some way or it might be a report based on analysis or result of the images.

4.4 Convolutional Neural Network

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It's also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image. [8]

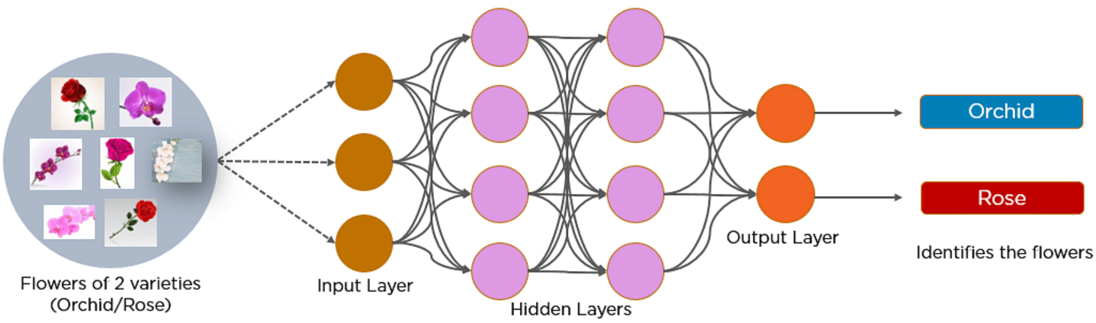


Figure 4.2 CNN

4.4.1 Working of Convolutional Neural Network

A convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer

4.4.1.1 Convolution Layer

In a Convolutional Neural Network (CNN), a convolutional layer is a type of layer that performs convolution operations on the input image or feature maps. The main purpose of a convolutional layer is to extract features from the input image or feature maps, which can then be used for further processing or classification.

A convolution operation involves sliding a small window called a filter or kernel over the input image or feature maps, performing element-wise multiplication between the filter and the corresponding input pixels, and then summing the results. The output of this operation is a single value that represents the presence of a particular feature or pattern at that location in the input.

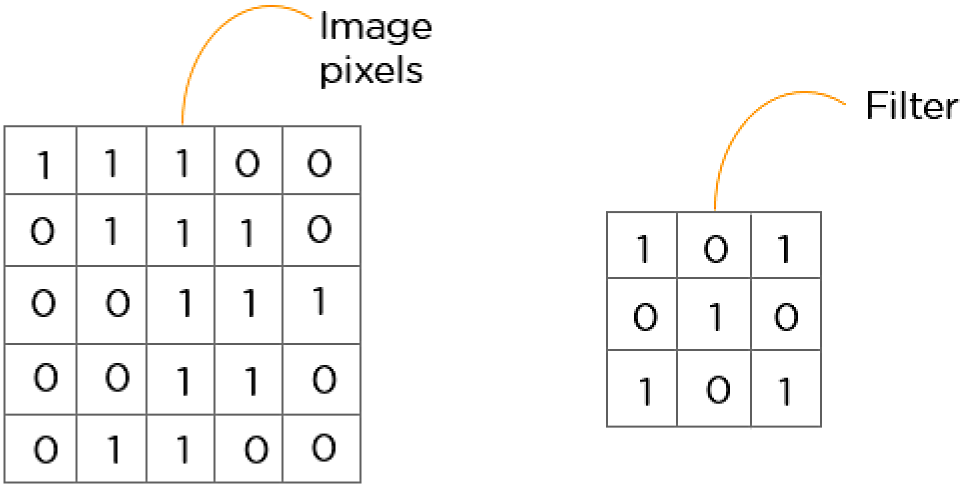


Figure 4.3 Convolutional Layer

4.4.1.2 ReLU layer

ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer.

ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map. Below is the graph of a ReLU function:

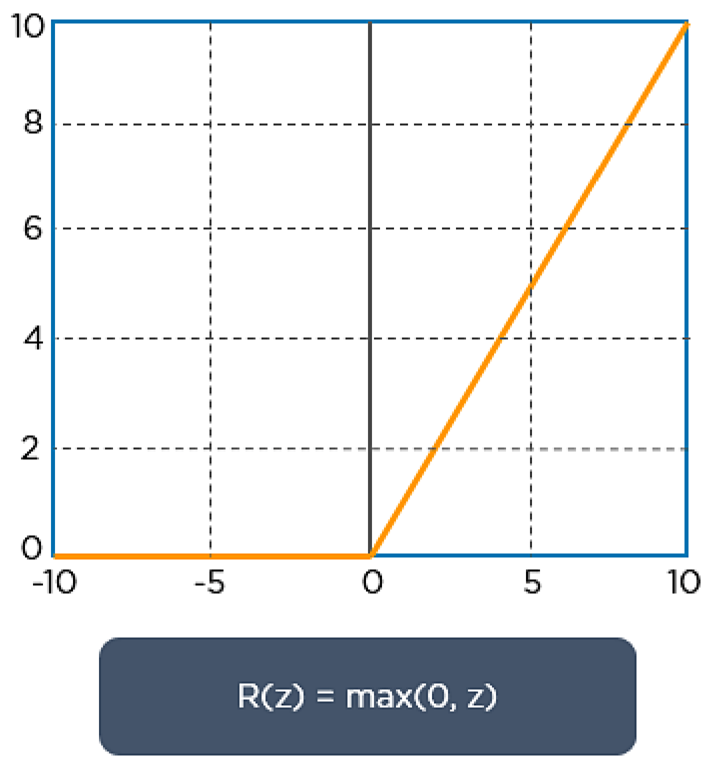


Figure 4.4 ReLU Layer

4.4.1.3 Pooling Layer

Pooling is a down-sampling operation that reduces the dimensionality of the feature map. The rectified feature map now goes through a pooling layer to generate a pooled feature map.

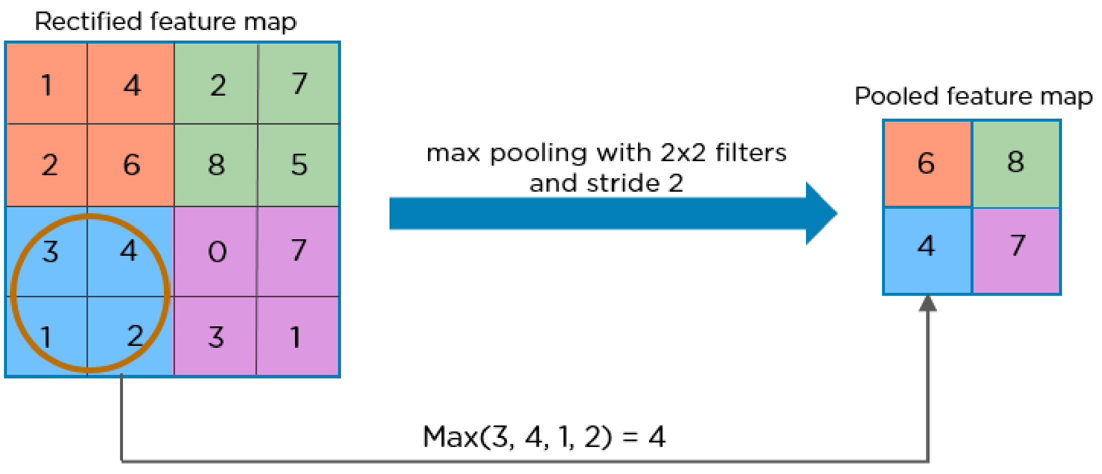


Figure 4.5 Pooling Layer

4.4.1.4 Flattening

Flattening is a process in a Convolutional Neural Network (CNN) where the output of the last convolutional layer is converted into a one-dimensional feature vector, which can then be fed into one or more fully connected layers for classification or regression.

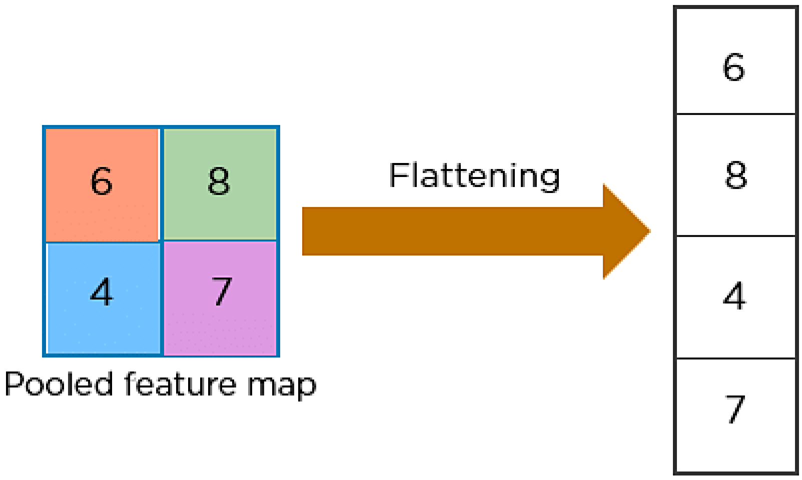


Figure 4.6 Flattening

4.4.1.5 Fully connected layer

This layer connects every neuron in the previous layer to every neuron in the next layer. This is similar to the layers in a traditional neural network, and is used to map the features learned by the earlier layers to the output classes.

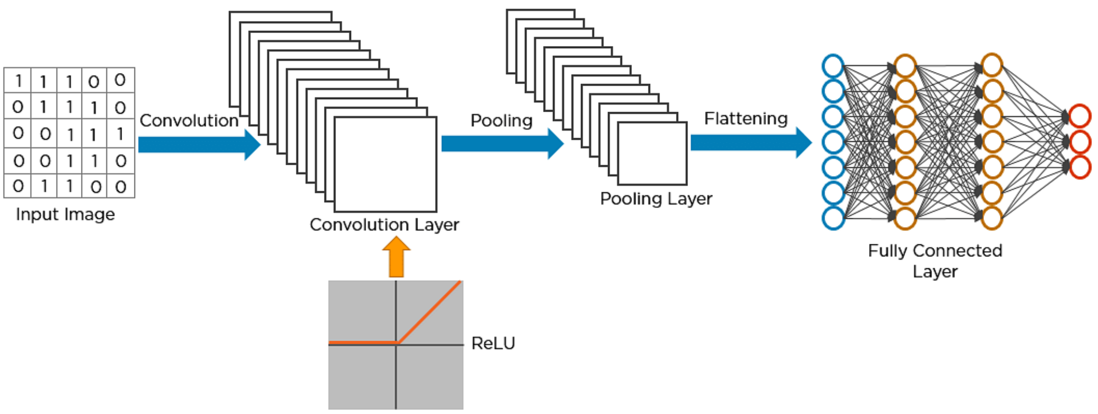


Figure 4.7 Fully Connected Layer

4.4.1.6 Output layer

This layer produces the final output of the network, which can be a probability distribution over a set of output classes, or a regression output.

4.5 Pesticide Recommendation

After identifying the type of disease present on the leaf, the model should recommend an appropriate pesticide to treat the disease.

4.6 User Interface

A user-friendly interface should be developed for the end-users to interact with the system. Users should be able to upload images of diseased leaves, and the system should output the disease and pesticide recommendations.

4.7 Advantages of Proposed System

There are several advantages of using Convolutional Neural Networks (CNNs) in multiple leaf disease detection and pesticide recommendation. Here are some of the key advantages:

1. Automatic feature extraction: One of the biggest advantages of CNNs is their ability to automatically learn and extract relevant features from images. This eliminates the need for manual feature engineering, which can be time-consuming and error-prone.
2. High accuracy: CNNs have been shown to achieve state-of-the-art performance on a wide range of computer vision tasks, including image classification and object detection. This makes them well-suited for multiple leaf disease detection, where accuracy is crucial for effective pest management.
3. Robustness to variations: CNNs are designed to be robust to variations in the input images, such as changes in lighting, orientation, and scale. This makes

them well-suited for leaf disease detection, where variations in leaf shape and color can make detection challenging.

4. Scalability: CNNs are highly scalable and can be trained on large datasets with millions of images. This makes them well-suited for multiple leaf disease detection, where large datasets are often required to capture the full range of leaf disease symptoms.
5. Real-time detection: CNNs can be designed to run in real-time, making them well-suited for use in automated pest management systems. This can help farmers detect leaf diseases early and take appropriate action to prevent their spread.
6. Pesticide recommendation: CNNs can also be used to recommend appropriate pesticides for specific leaf diseases. By combining multiple leaf disease detection with pesticide recommendation, farmers can take more targeted and effective action to prevent crop damage.

Overall, the use of CNNs in multiple leaf disease detection and pesticide recommendation can help improve the accuracy, speed, and effectiveness of pest management, leading to better crop yields and improved food security.

5 System Design

The system you are describing would involve the use of Convolutional Neural Networks (CNN) for detecting multiple leaf diseases in plants, and recommending appropriate pesticides for treatment. [9]

5.1 System Model

The system model for multiple leaf disease detection and pesticide recommendation using CNN (Convolutional Neural Network) can be developed in the following way:

1. Data collection: Collect a large dataset of leaf images with multiple diseases and healthy leaves. The dataset should include images of different crop plants.
2. Data preprocessing: Preprocess the collected data by resizing, normalizing, and augmenting the images. This ensures that the images are of the same size, format, and quality, which is important for the model's performance.
3. Model development: Develop a CNN model using a deep learning framework like TensorFlow, Keras, or PyTorch. The model should be designed to identify and classify multiple leaf diseases and recommend appropriate pesticides based on the disease type.
4. Model training: Train the developed model on the preprocessed dataset. The training process involves iteratively adjusting the model's weights and biases to minimize the error between the predicted output and the actual output. This process requires a large amount of computing power and time.
5. Model validation: Validate the trained model using a separate dataset of leaf images to evaluate its accuracy and performance. This dataset should be different from the training dataset to avoid overfitting.

6. Deployment: Deploy the trained model as a web application or mobile app for use by farmers or agricultural experts. The application should allow users to upload images of diseased leaves, and the model will provide a diagnosis and recommend the appropriate pesticide.

Overall, the system model will use deep learning techniques to automate the process of leaf disease detection and pesticide recommendation, which will help farmers save time and improve their crop yield. The accuracy of the model can be improved by continuously updating the dataset and retraining the model with new data. [10]

5.2 System Architecture

The following figure represents the architecture of the proposed system for the detection of multiple leaf diseases. It consists of different phases namely pre-processing, feature extractions, featured values, classifier and detect disease and provide treatment. The CNN architecture is a layered approach in which the layers are processed sequentially.

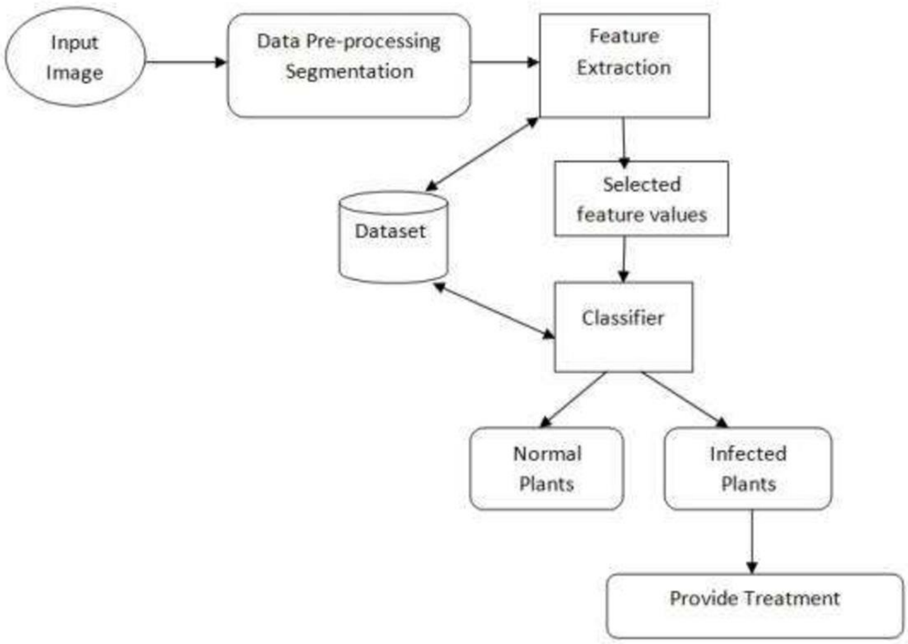


Figure 5.1 Architecture of Proposed System

5.3 Uml Diagrams

UML (Unified Modeling Language) diagrams are a standardized way of representing object-oriented software design. UML diagrams help developers, analysts, and other stakeholders to visualize, understand, and communicate complex systems and processes.

5.3.1 Usecase Diagram

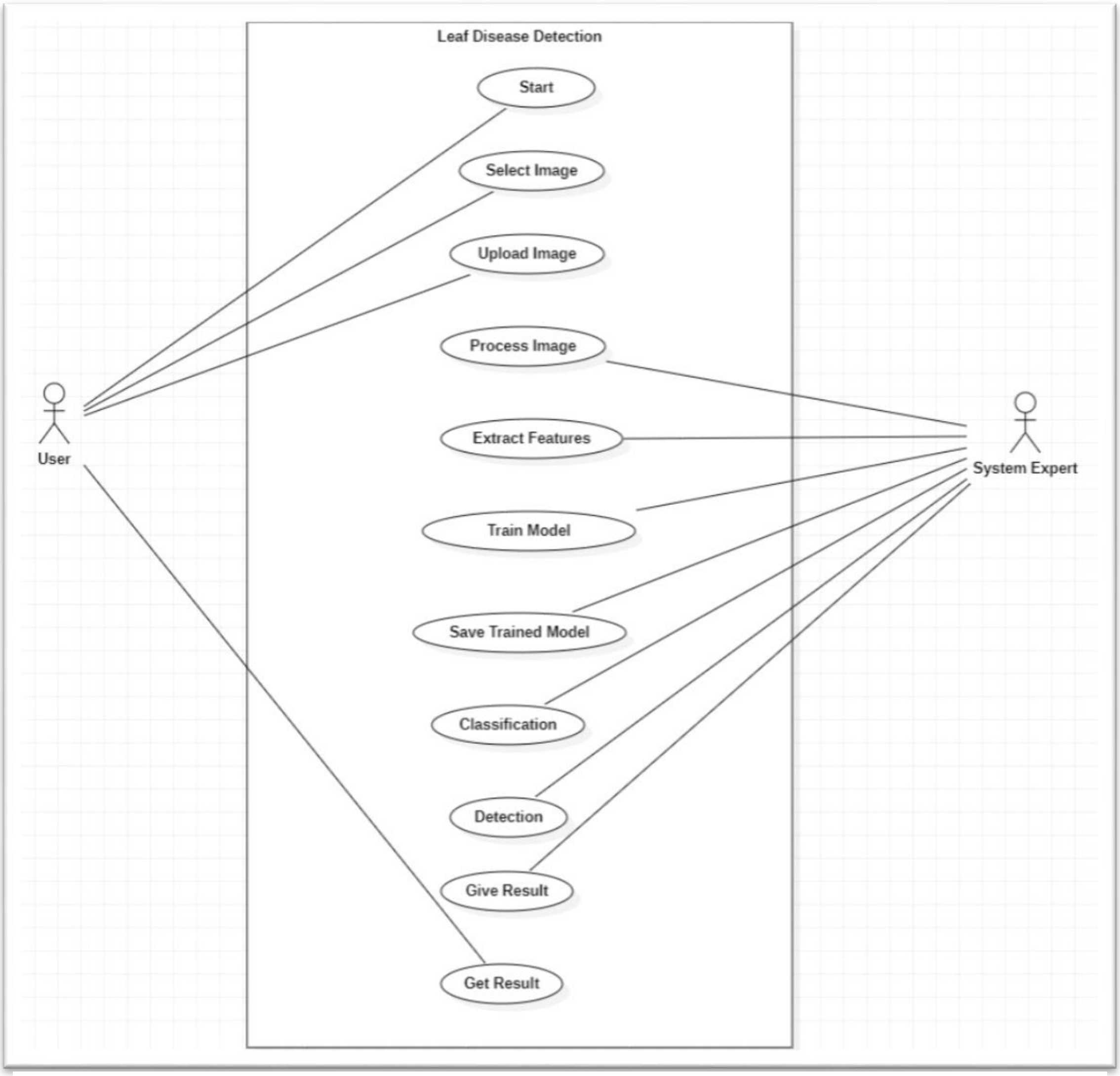


Figure 5.2 Usecase Diagram of Proposed System

5.3.2 Activity Diagram

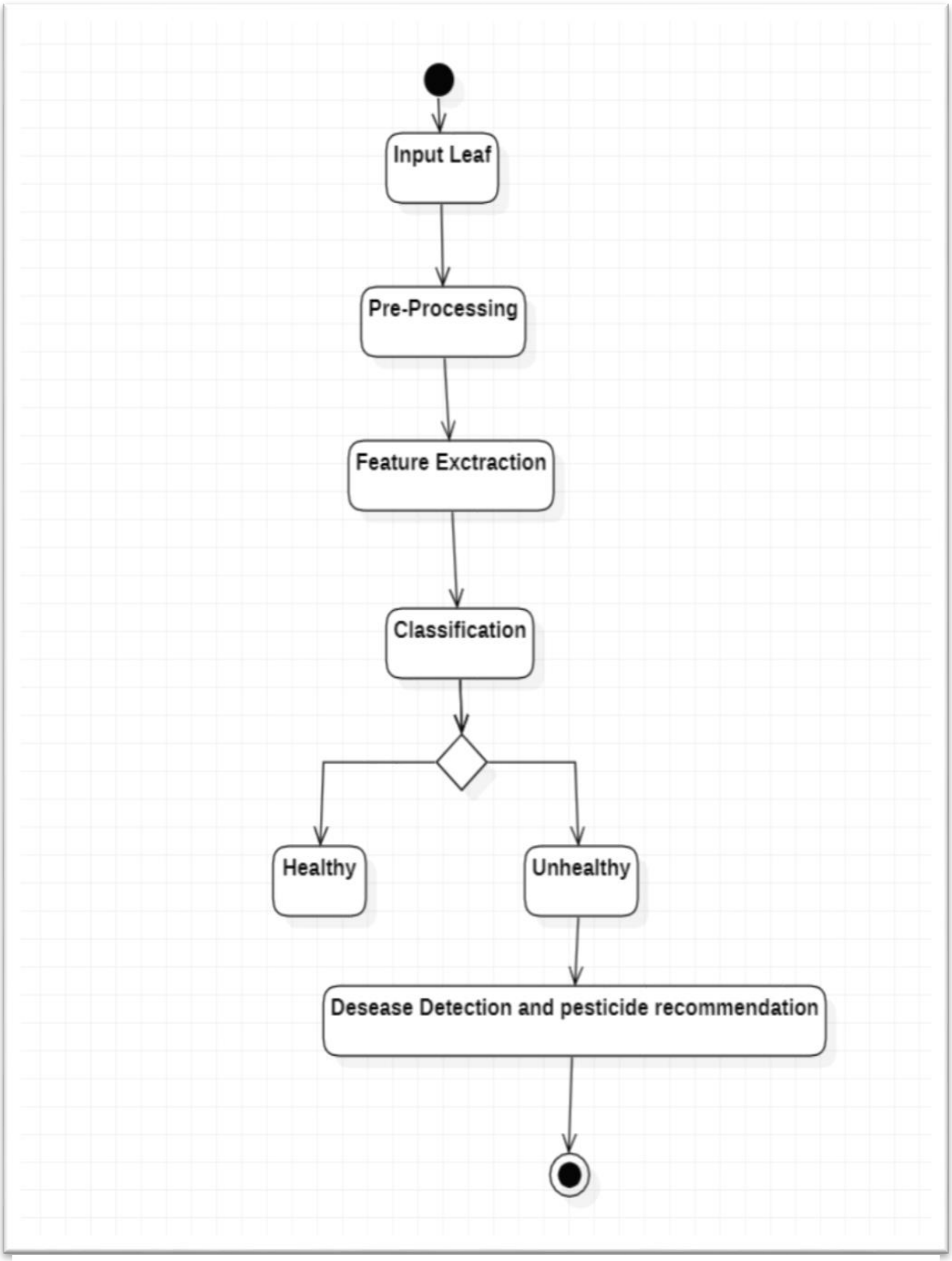


Figure 5.3 Activity Diagram of Proposed System

6 Implementation

Multiple leaf disease detection and pesticide recommendation using CNN involves a process where a computer vision model is trained using Convolutional Neural Networks (CNNs) to identify the various types of plant diseases on the leaves. Once the disease is identified, the model recommends the appropriate pesticide to treat the disease. The working of the system involves several steps such as Data Collection, Preprocessing, Training the CNN model, Disease Detection, Pesticide Recommendation, Deployment. [2]

6.1 Software Requirements

1. Operating System : Windows 7 and above.
2. Programing language : Python 2.7 and above.
3. Platform : Python IDLE
4. Supporting libraries : Tensorflow , Keras etc.
5. Data Set : Pictures of leafs

6.2 Hardware Requirements

1. Processor : Intel or AMD Ryzen.
2. RAM : A minimum of 4 GB of RAM.
3. Hard disk : A minimum of 20 GB of available space.

6.3 Libraries

1. **Streamlit:** Streamlit is an open-source Python library used for building interactive data science and machine learning web applications. It allows users to create beautiful and responsive data-driven apps with just a few lines of Python code.

Streamlit makes it easy to create and share interactive visualizations, dashboards, and other data-focused applications. It supports a wide range of data visualization and plotting libraries, including Matplotlib, Plotly, and Altair, among others.

2. **Streamlit_lottie:** Streamlit Lottie is a Streamlit component that allows you to easily add Lottie animations to your Streamlit app. Lottie is a library for adding animations to web and mobile applications using JSON files, which are small, lightweight, and can be easily embedded into a web page. With Streamlit Lottie, you can add Lottie animations to your Streamlit app by simply providing the path to your Lottie JSON file.
3. **Streamlit_option_menu:** Streamlit Option Menu is a Streamlit widget that allows you to create a dropdown menu that lets users select from a list of options. The streamlit.option_menu function takes a label for the widget and a list of options as arguments. The selected option is returned as the output of the function.
4. **Tensorflow:** TensorFlow is a software tool of deep learning. It is an AI library that allows developers to create large-scale multi-layered neural networks. It works on the basis of data flow graphs that have nodes and edges.
5. **Numpy:** NumPy(Numerical Python) is an open-source library of Python. It is designed to perform complex mathematical, image processing, quantum computing, and statistical operations, etc., on matrices and multidimensional arrays.
6. **Opencv_python:** OpenCV-Python is an open-source computer vision and machine learning software library that provides a wide range of tools and

algorithms for image and video processing. It is a Python wrapper for the original OpenCV library, which was written in C++.

7. **Pandas:** Pandas is an open-source Python library for data manipulation and analysis. It provides data structures for efficiently storing and manipulating large datasets, as well as tools for data cleaning, filtering, grouping, merging, reshaping, and visualization.
8. **Pillow:** Pillow is an open-source Python library that provides tools for working with image files. It is a fork of the Python Imaging Library (PIL) and provides more features and improvements over PIL. Pillow provides several classes and functions for opening, manipulating, and saving various image file formats such as JPEG, PNG, BMP, GIF, and more. You can use Pillow to resize, crop, rotate, flip, and filter images. You can also add text and graphics to images, create thumbnails, and convert between different image modes and color spaces.
9. **Requests:** Requests is an open-source Python library that provides tools for making HTTP requests. It allows you to send HTTP/1.1 requests, including GET, POST, PUT, DELETE, and more, and receive responses from web servers.Requests provides a simple and easy-to-use API that abstracts away the complexities of making HTTP requests. You can use it to send requests with various headers, parameters, and payloads, and to handle responses in various formats such as JSON, XML, and HTML.

6.4 Code Modules

For implementing the leaf disease detection the required source code is:

6.4.1 Download Dataset

```
od.download('https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset')
# sdlpkmrmkkml
# 91b742aa9b01dc96ac5940ea43212cdc
```

6.4.2 Defining path and identifying classes

```
data_dir = pathlib.Path("/content/plantvillage-dataset/plantvillage dataset/color")
train='/content/plantvillage-dataset/plantvillage dataset/color'
dataset_path_train = os.listdir(data_dir)
print (dataset_path_train)
print("Types of classes labels found: ", len(dataset_path_train))
```

6.4.3 Resizing image

```
image_count_train = len(list(data_dir.glob('*/*.JPG')))
print("The number of Train data:",image_count_train)
batch_size = 32
img_height = 224
img_width = 224
```

6.4.4 Training the data set

```
train_ds = image_dataset_from_directory(data_dir, seed = 123,
image_size=(img_height, img_width),
 validation_split=0.2,subset='training')
```

6.4.5 Validating Data

```
val_ds=image_dataset_from_directory(data_dir,seed=123,image_si
ze=(img_height,img_width),validation_split=0.2,subset='validat
ion')
```

6.4.6 Testing

```
test=image_dataset_from_directory(data_dir,seed=123,image_size
=(img_height, img_width),validation_split=None)
class_names = train_ds.class_names
print(len(class_names))
print( class_names)
for image_batch, labels_batch in train_ds.take(1):
    print(image_batch.shape)
    print(labels_batch.shape)
```

6.4.7 Building CNN Model

```
num_classes = 38
model = Sequential()
model.add(layers.experimental.preprocessing.Rescaling(1./255,
input_shape=(img_height, img_width, 3)))
model.add(Conv2D(16, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D())

model.add(Conv2D(32, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D())

model.add(Conv2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D())
model.add(Dropout(0.15))
```

```
model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dense(num_classes))
```

6.4.8 Compile Code

```
model.compile(optimizer='adam',
         loss=SparseCategoricalCrossentropy(from_logits=True),
          metrics=['accuracy'])
```

6.4.9 Save Modal

```
from tensorflow import expand_dims, newaxis

def predict(model, img):

    img_array = img.numpy()
    img_array = expand_dims(img_array, 0)

    predictions = model.predict(img_array)

    predicted_class = labels[np.argmax(predictions[0])]
    confidence = round( (np.max(predictions[0])), 2)

    return predicted_class, confidence

model.save("Model.h5")
```

6.4.10 Main Interface

```
# importing libraries
import streamlit as st
import numpy as np
from streamlit_option_menu import option_menu
from streamlit_lottie import st_lottie
import requests
from tensorflow import keras
from PIL import Image
import cv2
import os

file = open('pesticides.txt', 'r')
text = file.read()

labels     =     ['Apple___Apple_scab',     'Apple___Black_rot',
'Apple___Cedar_apple_rust',                      'Apple___healthy',
'Blueberry___healthy',
'Cherry_(including_sour)___Powdery_mildew',
'Cherry_(including_sour)___healthy',
'Corn_(maize)___Cercospora_leaf_spot          Gray_leaf_spot',
'Corn_(maize)___Common_rust_',
'Corn_(maize)___Northern_Leaf_Blight',
'Corn_(maize)___healthy',                      'Grape___Black_rot',
'Grape___Esca_(Black_Measles)',
'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
'Grape___healthy',  'Orange___Haunglongbing_(Citrus_greening)',
'Peach___Bacterial_spot',                        'Peach___healthy',
'Pepper,_bell___Bacterial_spot',      'Pepper,_bell___healthy',
'Potato___Early_blight',                      'Potato___Late_blight',
'Potato___healthy', 'Raspberry___healthy', 'Soybean___healthy',
'Squash___Powdery_mildew',          'Strawberry___Leaf_scorch',
'Strawberry___healthy',               'Tomato___Bacterial_spot',
'Tomato___Early_blight',                      'Tomato___Late_blight',
'Tomato___Leaf_Mold',            'Tomato___Septoria_leaf_spot',
'Tomato___Spider_mites               Two-spotted_spider_mite',
```

```
'Tomato___Target_Spot',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
'Tomato___Tomato_mosaic_virus', 'Tomato___healthy']

model = keras.models.load_model('Model.h5')

dic = {}

dic['Tomato___Bacterial_spot'] = 'BLUE COPPER , UTHANE M-45'
dic['Tomato___Early_blight'] =  "AMISTAR TOP,NATIVO"
dic['Tomato___Leaf_Mold'] = 'INDOFIL Z-78'
dic['Tomato___Late_blight'] = 'FOLIO GOLD,ABIC'
dic['Tomato___Spider_mites Two-spotted_spider_mite'] = 'OBERON'
dic['Tomato___Target_Spot'] = 'M-45INDOFIL,COPPEROXYCHLORIDE'
dic['Tomato___Tomato_Yellow_Leaf_Curl_Virus'] =
'CONFIDER,BENEVIA'
dic['Tomato___Septoria_leaf_spot'] = 'CHLOROTHALONIL,MANCOZEB'
dic['Tomato___Tomato_mosaic_virus'] = 'RAZE,FOLIOS'
dic['Pepper,_bell___Bacterial_spot'] = 'FOLIAR
SPRAY,CAREBENDAZIM 12.0% WP'
dic['Strawberry___Leaf_scorch'] = 'TRICHOCARE,MALATHION 50'
dic['Potato___Late_blight'] = 'MANCOZEB, DIMETHOMORPH 20.27%'
dic['Potato___Early_blight'] = 'KNOCKOUT NANO,PROPI'
dic['Squash___Powdery_mildew'] = 'BAVISTIN,DHANUSTIN'
dic['Grape___Black_rot'] = 'MANCOZEB'
dic['Grape___Esca_(Black_Measles)'] = 'CUSTODIA'
dic['Grape___Leaf_blight_(Isariopsis_Leaf_Spot)'] = 'CURZATE
M8'
dic['Cherry_(including_sour)___Powdery_mildew'] =
'PROPICONAZOLE'
dic['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot'] =
'ZEB M- 45,AZOXYSTROBIN'
dic['Corn_(maize)___Common_rust_'] = 'MACOBAN M-45,ABIC'
dic['Corn_(maize)___Northern_Leaf_Blight'] = 'INDOFIL Z-
78,MOUNT -45'
dic['Apple___Cedar_apple_rust'] = 'LIQUID COPPER'
dic['Apple___Black_rot'] = 'CAPTAN,SULFUR'
```

```
dic['Apple___Apple_scab'] = 'MYCLOBUTANIL'
dic['Orange___Haunglongbing_(Citrus_greening)']='ZINCO'
dic['Peach___Bacterial_spot']='FURY,RENO'

dic1 = {}

dic1['BLUE COPPER'] = 'https://amzn.eu/d/56MZZBN'
dic1['UTHANE M-45'] = 'https://amzn.eu/d/erW82yE'
dic1['AMISTAR TOP'] = 'https://amzn.eu/d/c7WcbcR'
dic1['NATIVO'] = 'https://amzn.eu/d/hCIfO8B'
dic1['INDOFIL Z-78'] = 'https://amzn.eu/d/3ZQZQZQ'
dic1['FOLIO GOLD'] = 'https://amzn.eu/d/4nQxDsm'
dic1['ABIC'] = 'https://amzn.eu/d/5biBSj7'
dic1['OBERON'] = 'https://amzn.eu/d/5AElS7L'
dic1['M-45INDOFIL'] = 'https://amzn.eu/d/0JLv5q2'
dic1['COPPEROXYCHLORIDE'] = 'https://amzn.eu/d/3FVXnre'
dic1['CONFIDER'] = 'https://amzn.eu/d/7X00LEN'
dic1['BENEVIA'] = 'https://amzn.eu/d/dY8OilX'
dic1['CHLOROTHALONIL'] = 'https://amzn.eu/d/dKViMd1'
dic1['MANCOZEB'] = 'https://amzn.eu/d/3ChkiJf'
dic1['RAZE'] = 'https://amzn.eu/d/12rLJmW'
dic1['FOLIOS'] = 'https://amzn.eu/d/7TpuIy5'
dic1['FOLIAR SPRAY'] = 'https://amzn.eu/d/9LK8od2'
dic1['CAREBENDAZIM 12.0% WP'] = 'https://www.amazon.in/Care-500-Grams-Carbendazim-Fungicide/dp/B08TB16VYG'
dic1['TRICHOCARE'] = 'https://amzn.eu/d/9VDEmZT'
dic1['MALATHION 50'] = 'https://amzn.eu/d/9VDEmZT'
dic1['DIMETHOMORPH 20.27%'] = 'https://amzn.eu/d/bTpVPYK'
dic1['KNOCKOUT NANO'] = 'https://amzn.eu/d/hiuacz3'
dic1['PROPI'] = 'https://amzn.eu/d/d3Pi6uH'
dic1['BAVISTIN'] = 'https://amzn.eu/d/4bRrr4q'
dic1['CUSTODIA'] = 'https://amzn.eu/d/ff3rmLi'
dic1['CURZATE M8'] = 'https://amzn.eu/d/01Jf9Ew'
dic1['PROPICONAZOLE'] = 'https://amzn.eu/d/0IVf3jp'
dic1['ZEB M- 45'] = 'https://amzn.eu/d/3Aom2UO'
dic1['AZOXYSTROBIN'] = 'https://amzn.eu/d/0uoaHoS'
dic1['MACOBAN M-45'] = 'https://amzn.eu/d/7vjgWvD'
```

```
dic1['INDOFIL Z-78'] = 'https://amzn.eu/d/276XfPD'
dic1['LIQUID COPPER'] = 'https://amzn.eu/d/1TrelaV'
dic1['CAPTAN'] = 'https://amzn.eu/d/13pz4g6'
dic1['SULFUR'] = 'https://amzn.eu/d/ibeppTH'
dic1['MYCLOBUTANIL'] =
'https://farmkey.in/product/index%E0%A4%87%E0%A4%82%E0%A4%A1%E0%A5%87%E0%A4%95%E0%A5%8D%E0%A4%B8'
dic1['ZINCO']='https://amzn.eu/d/6j6RAie'
dic1['FURY']='https://amzn.eu/d/ceU75ju'
dic1['RENO']='https://amzn.eu/d/j02A4b6'

def predict(img):
    img = np.asarray(img)
    img = cv2.resize(img, (224, 224))
    img = img.reshape(1, 224, 224, 3)
    prediction = model.predict(img)
    label = labels[np.argmax(prediction[0])]
    confidence = round( (np.max(prediction[0])), 2)
    return label, confidence



def load_lottieurl(url: str):
    r = requests.get(url)
    if r.status_code != 200:
        return None
    return r.json()

def main():

    # sidebar for navigation
    with st.sidebar:

        selected = option_menu('Leaf Disease Detector',

                               ['Home','Disease Detector'],
                               icons=['activity','heart',],
                               default_index=0)
```

```
    if selected == "Home":
        st.markdown("<h1  style='text-align:  center;  color:
#08e08a;'>Leaf Disease Detector</h1>", unsafe_allow_html=True)
        st.markdown("<h4 '>A web app to detect diseases in leafs
using    Deep    Learning    and    computer    vision</h4>",
unsafe_allow_html=True)
        lottie_hello                                          =
load_lottieurl("https://assets5.lottiefiles.com/packages/lf20_
1plcwvk5.json")
        st_lottie(
            lottie_hello,
            speed=1,
            reverse=False,
            loop=True,
            quality="High",
            #renderer="svg",
            height=400,
            width=-900,
            key=None,
        )
        st.markdown("<h4                        '>About:</h4>",
unsafe_allow_html=True)
        st.write("Food security for billions of people on earth
requires  minimizing  crop  damage  by  timely  detection  of
diseases.Developing  methods  for  detection  of  plant  diseases
serves the dual purpose of increasing crop yield and reducing
pesticide use without knowing about the proper disease. Along
with development of better crop varieties, disease detection is
thus paramount goal for achieving food security. The traditional
method of disease detection has been to use manual examination
by either farmers or experts, which can be time consuming and
costly, proving infeasible for millions of small and medium sized
farms around the world.")

        st.markdown("<h4                     '>Features:</h4>",
unsafe_allow_html=True)
```

```
        st.write("Easy Detection of diseases in leafs: Just need
to click and upload leaf image.")
        st.write("Fast and Accurate: Provides the disease with
high accuracy and fast")
        st.write("Cause and Solution of diseases: Provides the
cause and solution of the disease")
        st.write('Large  Plant  Support:  Supports  38  different
classes')

        # list of plants we can detect
        st.markdown("<h4  '>List of Plants we can predict</h4>",
unsafe_allow_html=True)
        st.write(labels)
        lotti                                                    =
load_lottieurl("https://assets5.lottiefiles.com/packages/lf20_
srcvuh0h.json")
        st_lottie(
            lotti,
            speed=1,
            reverse=False,
            loop=True,
            quality="High",
            #renderer="svg",
            height=400,
            width=-900,
            key=None,
        )

    elif selected == "Disease Detector":
        st.markdown("<h1   style='text-align:   center;   color:
#08e08a;'>Disease Detector</h1>", unsafe_allow_html=True)
        lot                                                      =
load_lottieurl('https://assets3.lottiefiles.com/packages/lf20_
0xbu1xfo.json')
        st_lottie(
            lot,
            speed=1,
```

```
            reverse=False,
            loop=True,
            quality="High",
            #renderer="svg",
            height=400,
            width=-900,
            key=None,
        )
        image  = st.file_uploader("Upload Image", type=['jpg',
'png', 'jpeg'])
        if image is not None:
            image = Image.open(image)
            st.image(image,      caption="Uploaded      Image",
use_column_width=True)
            st.write("")
            label, confidence = predict(image)
            st.write(f"**Predicted Class:** {label}")
            st.write(f"**Confidence:** {confidence}")
            st.write("")
            x = label
            if x in text:
                if text.split(x)[1].splitlines()[1] != '':
                    st.write(text.split(x)[1].splitlines()[1])
                    if text.split(x)[1].splitlines()[2] != '':

st.write(text.split(x)[1].splitlines()[2])
                        if  text.split(x)[1].splitlines()[3]  !=
'':

st.write(text.split(x)[1].splitlines()[3])
                            if text.split(x)[1].splitlines()[4]
!= '':

st.write(text.split(x)[1].splitlines()[4])
                                if
text.split(x)[1].splitlines()[5] != '':
```

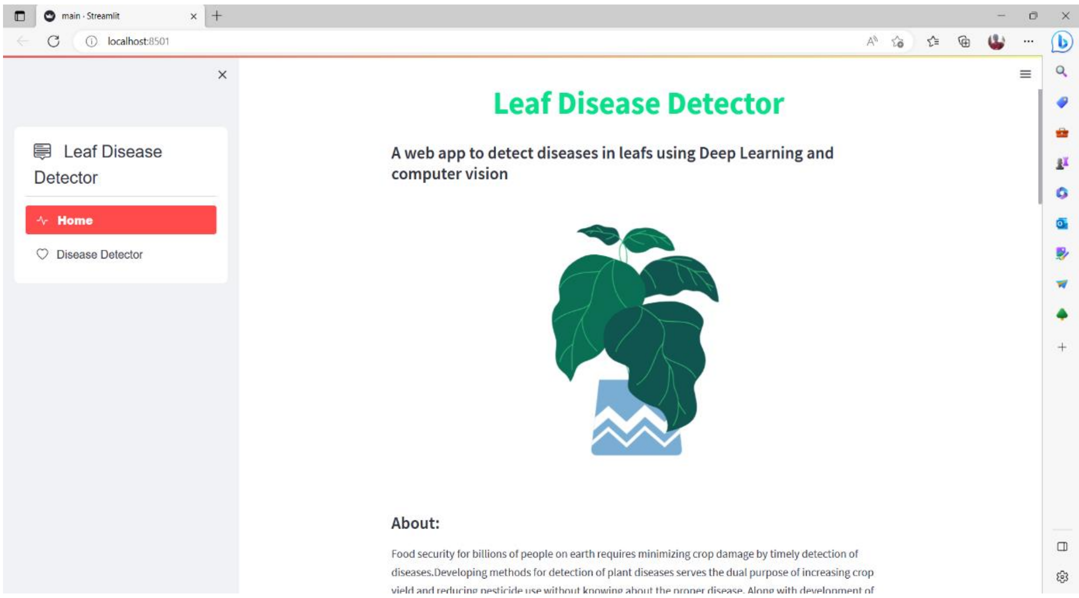
```
st.write(text.split(x)[1].splitlines()[5])
                                    if
text.split(x)[1].splitlines()[6] != '':

st.write(text.split(x)[1].splitlines()[6])
                                        if
text.split(x)[1].splitlines()[7] != '':

st.write(text.split(x)[1].splitlines()[7])
                                            if
text.split(x)[1].splitlines()[8] != '':

st.write(text.split(x)[1].splitlines()[8])
                                                if
text.split(x)[1].splitlines()[9] != '':

st.write(text.split(x)[1].splitlines()[9])
                                                    if
text.split(x)[1].splitlines()[10] != '':

st.write(text.split(x)[1].splitlines()[10])
                                                        if
text.split(x)[1].splitlines()[11] != '':

st.write(text.split(x)[1].splitlines()[11])
                                                            if
text.split(x)[1].splitlines()[12] != '':

st.write(text.split(x)[1].splitlines()[12])

if text.split(x)[1].splitlines()[13] != '':

st.write(text.split(x)[1].splitlines()[13])

            st.write("Suggested Pesticides:")
            pes = dic[x]
```

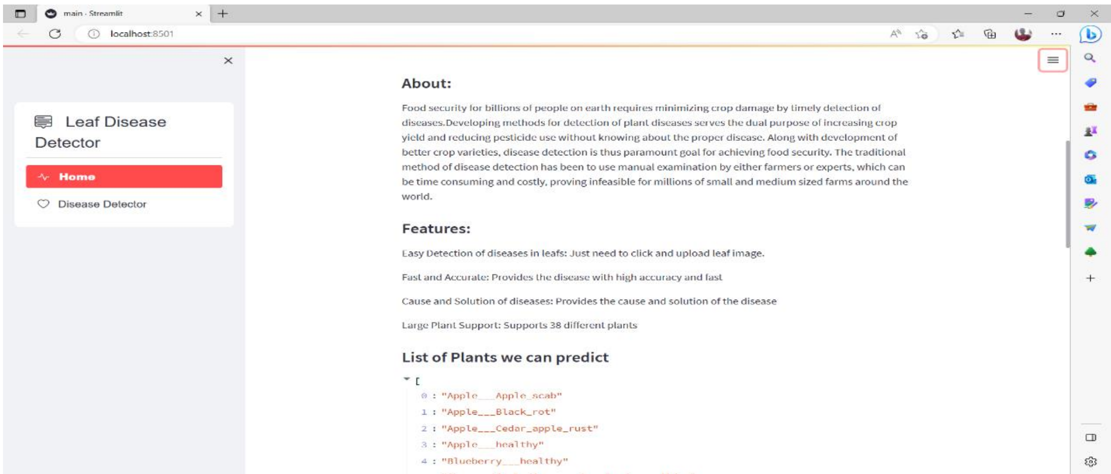
```
            st.write(pes)
            pes = pes.split(',')
            for i in pes:
                path = 'Pesticides/' + i + '.jpg'
                if os.path.exists(path):
                    st.write("Which looks like:")
                    st.image(path,                   caption=i,
use_column_width=True)
                    st.write("You can buy it from here:")
                    st.write(dic1[i])
                    pass
                else:
                    pass

if __name__ == "__main__":
    main()
```

7 Experimental Results

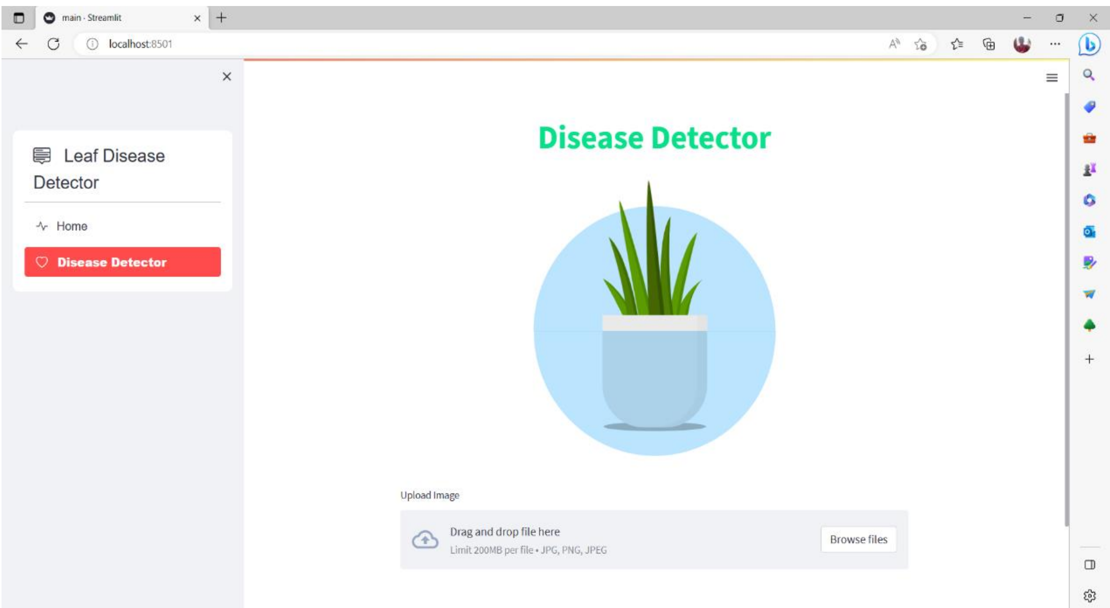
In general, experiments on leaf disease detection and pesticide recommendation using CNN involve collecting a dataset of images of leaves that have been affected by various diseases, as well as healthy leaves. The dataset is then split into training and testing sets, and the CNN model is trained on the training set to learn to recognize different types of diseases.

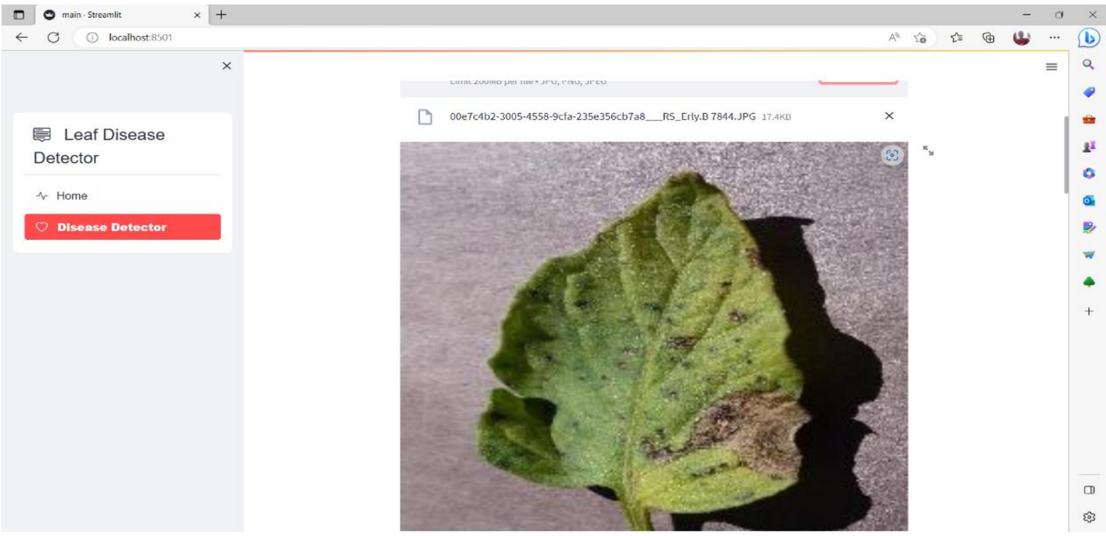
Once the model is trained, it is tested on the testing set to evaluate its accuracy in detecting diseases. If the model performs well in detecting diseases, it can then be used to recommend appropriate pesticides for the specific diseases detected.

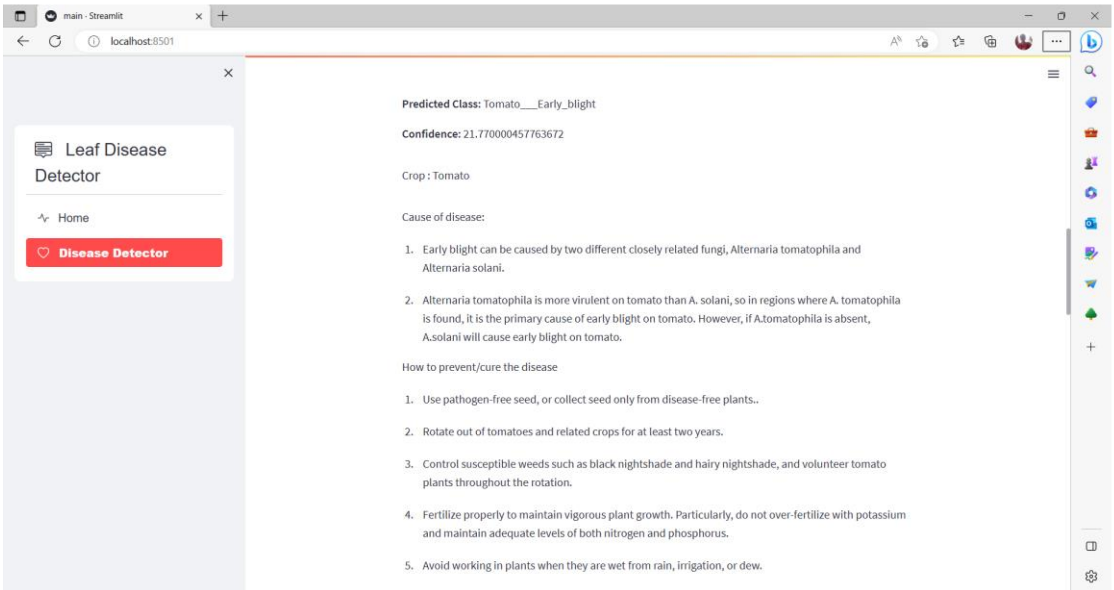



main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
About:
Food security for billions of people on earth requires minimizing crop damage by timely detection of diseases.Developing methods for detection of plant diseases serves the dual purpose of increasing crop yield and reducing pesticide use without knowing about the proper disease. Along with development of better crop varieties, disease detection is thus paramount goal for achieving food security. The traditional method of disease detection has been to use manual examination by either farmers or experts, which can be time consuming and costly, proving infeasible for millions of small and medium sized farms around the world.
Features:
Easy Detection of diseases in leafs: Just need to click and upload leaf image.
Fast and Accurate: Provides the disease with high accuracy and fast
Cause and Solution of diseases: Provides the cause and solution of the disease
Large Plant Support: Supports 38 different plants
List of Plants we can predict
0 : "Apple___Apple_scab"
1 : "Apple___Black_rot"
2 : "Apple___Cedar_apple_rust"
3 : "Apple___healthy"
4 : "Blueberry___healthy"

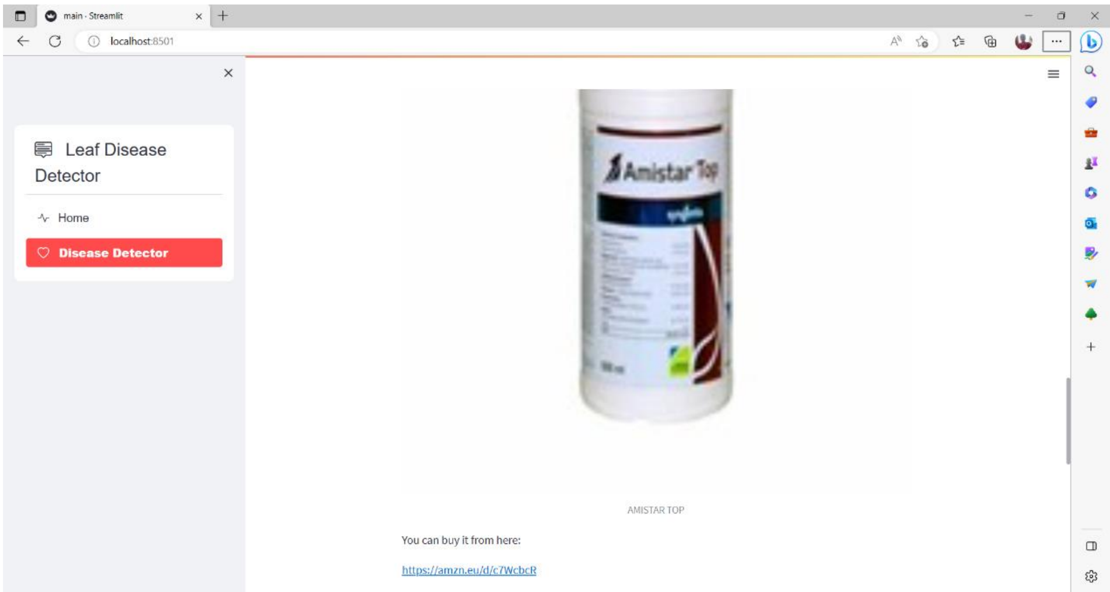

main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
14 : "Grape___healthy"
15 : "Orange___Haunglongbing_(Citrus_greening)"
16 : "Peach___Bacterial_spot"
17 : "Peach___healthy"
18 : "Pepper,_bell___Bacterial_spot"
19 : "Pepper,_bell___healthy"
20 : "Potato___Early_blight"
21 : "Potato___Late_blight"
22 : "Potato___healthy"
23 : "Raspberry___healthy"
24 : "Soybean___healthy"
25 : "Squash___Powdery_mildew"
26 : "Strawberry___Leaf_scorch"
27 : "Strawberry___healthy"
28 : "Tomato___Bacterial_spot"
29 : "Tomato___Early_blight"
30 : "Tomato___Late_blight"
31 : "Tomato___Leaf_Mold"
32 : "Tomato___Septoria_leaf_spot"
33 : "Tomato___Spider_mites Two-spotted_spider_mite"
34 : "Tomato___Target_Spot"
35 : "Tomato___Tomato_Yellow_Leaf_Curl_Virus"
36 : "Tomato___Tomato_mosaic_virus"
37 : "Tomato___healthy"


main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
Disease Detector
Upload Image
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files


main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
00e7c4b2-3005-4558-9cfa-235e356cb7a8___RS_Erly.B 7844.JPG 17.4KB


main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
Predicted Class: Tomato___Early_blight
Confidence: 21.770000457763672
Crop : Tomato
Cause of disease:
1. Early blight can be caused by two different closely related fungi, Alternaria tomatophila and Alternaria solani.
2. Alternaria tomatophila is more virulent on tomato than A. solani, so in regions where A. tomatophila is found, it is the primary cause of early blight on tomato. However, if A.tomatophila is absent, A.solani will cause early blight on tomato.
How to prevent/cure the disease
1. Use pathogen-free seed, or collect seed only from disease-free plants..
2. Rotate out of tomatoes and related crops for at least two years.
3. Control susceptible weeds such as black nightshade and hairy nightshade, and volunteer tomato plants throughout the rotation.
4. Fertilize properly to maintain vigorous plant growth. Particularly, do not over-fertilize with potassium and maintain adequate levels of both nitrogen and phosphorus.
5. Avoid working in plants when they are wet from rain, irrigation, or dew.


main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
Suggested Pesticides:
AMISTAR TOP,NATIVO
Which looks like:


main - Streamlit
localhost:8501
Leaf Disease Detector
Home
Disease Detector
Amistar Top
AMISTAR TOP
You can buy it from here:
https://amzn.eu/d/c7WcbcR

8 Comparisions

Both SVM (Support Vector Machines) and CNN (Convolutional Neural Networks) are machine learning algorithms that can be used for classification tasks such as disease detection in plants. However, the approach and the scope of each algorithm differ.

Single leaf disease detection using SVM involves training a machine learning model on a set of labeled images of healthy and diseased leaves. The model then learns to classify new, unseen images based on the features extracted from the image. SVM is a popular algorithm for this task as it can handle high-dimensional data well and has been shown to perform well on image classification tasks.

On the other hand, multiple leaf disease detection and pesticide recommendation using CNN involves training a deep learning model on a large dataset of labeled images of multiple plant diseases and pests. The CNN model can extract more complex features from the images compared to SVM, and it can learn to detect multiple diseases simultaneously. Additionally, the recommendation of pesticides can be included in this system, making it more comprehensive.

SVM is used in the existing system and it resulted in 83% accuracy. We are using CNN Algorithm which gives more accurate results than SVM. The CNN model resulted up to 91% accuracy which is 8% more accurate than SVM.

In summary, SVM is a suitable algorithm for single leaf disease detection, while CNN is more appropriate for multiple leaf disease detection and pesticide recommendation due to its ability to extract complex features from images and classify multiple diseases simultaneously.

9 Conclusions

The use of Convolutional Neural Networks (CNNs) for multiple leaf disease detection and pesticide recommendation has shown promising results.

By training the CNN on a large dataset of healthy and diseased plant images, the model can accurately classify and identify the type of disease present on the leaves. This allows for early detection of the disease and the timely application of pesticides, reducing crop losses and improving yields.

The CNN can also be used to recommend specific pesticides based on the type of disease detected. This personalized approach can lead to more effective pest management and reduced pesticide use, resulting in cost savings and environmental benefits.

Overall, the use of CNNs for multiple leaf disease detection and pesticide recommendation holds great potential for improving agricultural productivity and sustainability. However, further research and testing are needed to optimize the model's performance and ensure its practicality and scalability for real-world applications.

10 Future Scope

The future scope for multiple leaf disease detection and pesticide recommendation using CNN (Convolutional Neural Network) is very promising. As the world's population grows, the demand for food increases, and the need for crop protection against diseases and pests becomes more critical. Thus, utilizing deep learning techniques such as CNN can provide farmers with a reliable and cost-effective way to detect plant diseases and make informed decisions about pesticide application.

Currently, the use of CNN for plant disease detection and pesticide recommendation is limited to a few crops and diseases. In the future, more crops and diseases can be added to the model, allowing it to be used on a wider scale.Currently, the proposed work only works on detection different leaf diseases and recommend pesticides. In the future,Seviarity of the leaf disease can also be found to make farmers easy and efficient.Autonomous agricultural machinery, such as drones and robots, can be integrated with CNN technology for plant disease detection and pesticide recommendation. This will enable farmers to monitor their crops without physically being present in the field, saving time and increasing efficiency.

11 References

[1] a. S. N. P. P. Narvekar, "“Novel algorithm for grape leaf diseases detection”," P. Narvekar, and S. N. Patil, “Novel algorithm for gInternational Journal of Engineering Research and General Science Volume 3, Issue 1, pp. pp.no 1240-1244, 2015.

[2] M. A. J. Hossain, "Leaf shape identification based plant biometrics," J. Hossain, M.A. Amin, LeIn Proceedings of the 2010 13th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, pp. J. Hossain, M.A. Amin, Leaf shape identification based plant biometrics. In Proceedings of the 2010 13th Inpp. 458–463, 2010.

[3] V. S., A. Singla, C. Ghanshyam, P. Kapur and S. Gupta, "Plant leaf imaging technique for agronomy.," S., V.D.; Singla, A.; Ghanshyam, C.; Kapur, P.; Gupta, S. Plant leaf imaging technique for agroInternational Conference on Image Information Processing, Shimla, India, , p. pp. 1–5., 2011.

[4] R. K. a. M. Kaur, "An Adaptive Model to Classify Plant Diseases Detection using KNN," R. Kaur and M. Kaur” An Adaptive Model to Classify Plant DiInternational Journal of Trend in Scientific Research and Development (IJTSRD) ISSN, pp. pp. 12-14, 2020.

[5] P. J. a. R. Kumar2, "FEATURE EXTRACTION OF DISEASED LEAF IMAGES," Journal of Signal and Image Processing ISSN: 0976-8882 & E-ISSN: 0976-8890, Volume 3, Issue 1, pp. pp.-60-63. , 2012.

[6] D. P. H. a. M. S. Sharada P Mohanty, "“Using deep learning for image-based plant disease detection”," In: Frontiers in plant science 7 , p. p. 1419., 2016.

[7] T. A. V.. M. J, "A Survey on Plant Disease Identification,"] V.. M. J [1], Thamizharasi. A [2] “A Survey on Plant International Journal of Computer Science Trends and Technology (IJCST), p. Volume 3 Issue 6, 2015.

[8] A.-K. U. S. E.-C. O. H.-W. D. a. L. P. T. Rumpf, "T. Rumpf, A.-K.Mahlein, U. Steiner, E.-C. Oerk“Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," T. Rumpf, A.-K.Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Pl'umer, “Early detection and classificatioComputers and Electronics in Agriculture, pp. T. Rumpf, A.-K.Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Pl'umer, “Early detection and classification ovol. 74, no. 1, 2021.

[9] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, p. pp. 311–318., 2017.

[10] Q. L. a. L. G. J. Chen, ""Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model"," Symmetry, vol. 11,no. 3 Available: 10.3390/sym11030343. , p. p. 343, 2019.