



# ENHANCING AGRICULTURE WITH MACHINE LEARNING: LEAF DISEASE DIAGNOSIS

<sup>1</sup>Chinnapaga Ganesh, <sup>2</sup>Dr. K Shanthi Sree

<sup>1</sup>Student, <sup>2</sup>Professor

<sup>1,2</sup>Department of Information Technology,

<sup>1,2</sup>JNTUH University College of Engineering Science & Technology, Hyderabad, Telangana, India

**Abstract:** One area that significantly affects people's quality of life and financial situation is agriculture. Poor management causes agricultural goods to be lost. This method uses a deep neural network, an alternative to a regular neural network, to identify leaf diseases. This makes it simple to identify leaf disease. The user can upload a picture of a leaf first, and the image will be uploaded on the screen. Next, click the button to analyse the image. A leaf's state of health can be determined by the disease and shown. The leaf image shows the presence of the illness. Utilizing an accessible dataset of 1500 images of solid and ill plants, this work characterizes crop species and determines the disease status of three different groups using deep convolutional systems and semi-supervised approaches.

**Keywords:** Image analysis, Semi-Supervised Approaches, Crop Species Classification, Disease Identification.

## I. INTRODUCTION

One of the most important jobs in the world is agriculture. Given that food is a basic necessity for all life on Earth, it plays a crucial role. Working on the nature of horticulture merchandise has therefore become essential. Accurately handling these yields from the start is fundamental. Plant life is segmented into multiple stages. Plant life is segmented into multiple stages. Included are all aspects of soil preparation, planting, adding manures and compost, water system techniques, determining the level of infection, using pesticides, and gathering produce. For example, between 20 and 40 percent of crop production accidents are caused by animals, weeds, nematodes, insects, and diseases. According to certain assessments, crop contaminations result in average result misfortunes of 42% for the major food crops. Due to their interference with photosynthesis, the process by which plants produce energy, leaf spot diseases harm trees and shrubs. As a result, the initial sickness prediction is simplistic. Crop diseases can occasionally eliminate the efficiency of an entire yield. Consequently, in order to truly control illnesses, ranchers need swiftly learn everything they can about disease modification. Owing to the increasing desire for food and food products among the modern population, horticultural frameworks have adopted a broad approach to handling the usage of manures for development purposes. This approach focuses on early detection of leaf diseases, reducing the likelihood that the entire plant will succumb to disease. There are two conventional techniques for identifying illnesses: manual evaluation of a leaf and disease prediction. However, this method does not allow a rancher to identify the precise ailment. Thus, it may be possible to identify plant leaf infection by knowing the proper sick picture handling techniques. These cutting-edge techniques use cutting-edge innovation to provide precise outcomes.

## II. EXISTING SYSTEM

In current systems, effective techniques including texture, colour analysis, and K-mean clustering are used to identify diseases. It uses characteristics of texture and colour that are typical of both normal and affected areas to distinguish between different types of agriculture. Other techniques that are employed include support vector machines, back propagation neural networks, extended regression neural networks, and conventional multiple regression (SVM). The link between environmental factors and illness level was more accurately represented by the SVM-based regression technique, which may be useful for managing diseases.

### 2.1 Disadvantages of the Existing System

The findings demonstrated that applying an SVM-based regression technique can be useful for the identification of plant diseases in addition to improving the description of the relationship between environmental conditions and disease level.

Current methods, including k-means and SVM, are ineffective; they require a lot of time to analyze and produce forecasts with poor accuracy.

## III. PROPOSED SYSTEM

In this research, we are using the CNN model and machine learning to forecast plant leaf disease. We trained a CNN model using a plant village dataset that we obtained from Kaggle and saved the model. Users can upload images and verify their sickness using an easy-to-use graphical user interface (GUI).

### 3.1. Proposed Architecture:

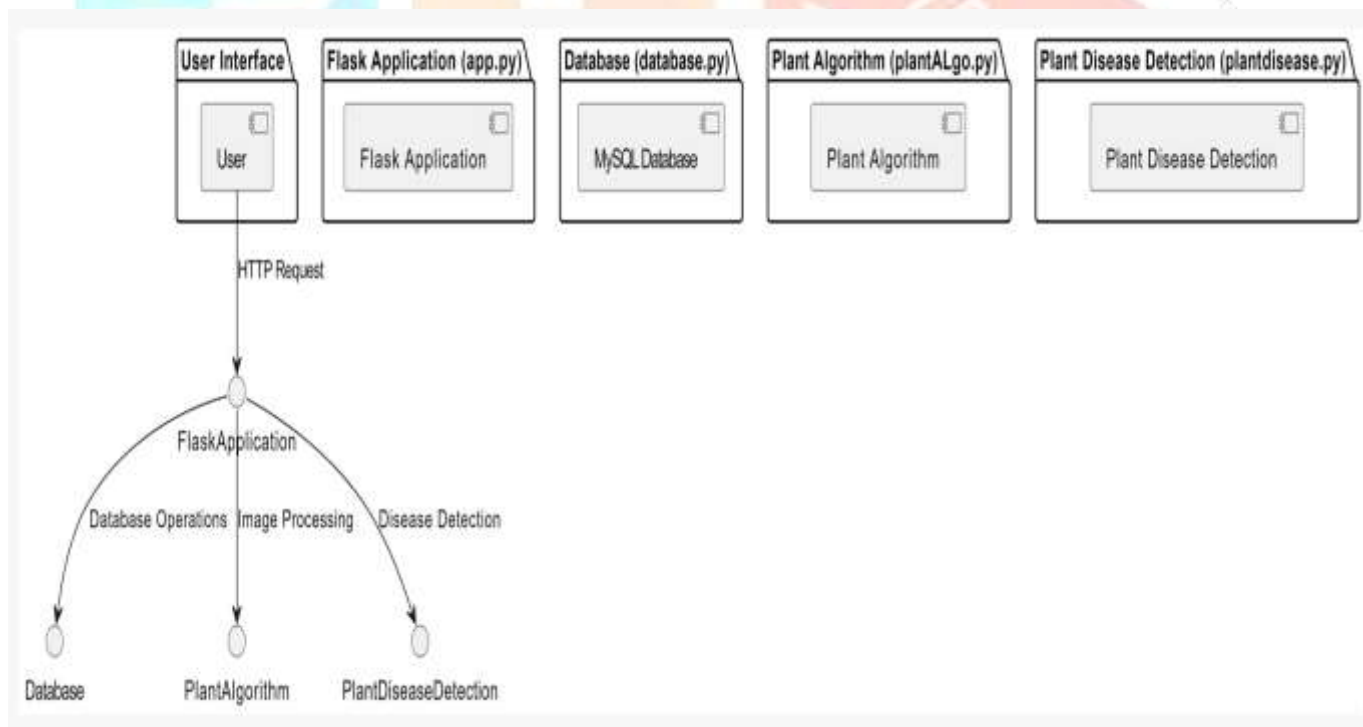


Fig 3.1: Proposed Architecture

### 3.2 Advantages of Proposed System:

The agricultural industry and farmers' interests were taken into consideration when developing the recommended strategy. The suggested approach provides both a diagnosis and a treatment plan for plant diseases. Understanding the disease and how to treat it thoroughly can help to improve the health of the plant.

## IV. IMPLEMENTATION

The first stage in our proposed model is data collection using Kaggle (a dataset source). The disease will then be detected using a combination of processing and feature extraction approaches.

### 4.1 Data Collection

Photos of numerous plant diseases are included in the collection. This system includes cash crops, fruits, cereals, vegetables, rice, wheat, bananas, guava, sugarcane, cotton, potato, carrot, chilly, and brinjal. For the crops mentioned above, pictures of healthy and diseased leaves as well as everything in between were collected from a range of sources, such as the internet or simple camera equipment photos.

### 4.2 Image Pre-Processing

Visualize the justification and growth An essential component of many photo search and recovery tools is image comment, which is the assignment of naturally occurring depiction words for an image. But in this architecture, we use a bounding box and class to physically comment on the area of each image that depicts the illness. Some diseases may seem similar depending on the level of contamination.

### 4.3 Image Analysis

Our system's main goal is to locate and detect instances of class sickness in images. It is necessary to correctly identify the object and determine its class. We improve the idea of the object detection framework by identifying diseases in photos by utilizing several feature extractors.

### 4.4 Analysis of the Implementation

The provided code encompasses a Flask web application that integrates with a MySQL database and utilizes machine learning models for classifying plant diseases from images. This analysis covers the following components:

- 1 Machine Learning Model Training
- 2 Machine Learning Model Integration
- 3 Database Interaction
- 4 Flask Web Application

#### 1. Machine Learning Model Training (plant\_ALgo.py)

##### Data Preparation:

- Images are loaded from the dataset directory and augmented using ImageDataGenerator to improve the model's robustness.
- Data is split into training and validation sets.

##### Model Definition:

- A VGG19 model pre-trained on ImageNet is used for transfer learning.
- The top layer of the VGG19 model is replaced with a custom classifier tailored for plant disease classification.

##### Model Training:

- Categorical cross-entropy loss and an Adam optimizer are used in the compilation of the model.
- Training is conducted with early stopping to prevent overfitting, using a custom callback to stop training if the loss falls below a certain threshold.

##### Model Evaluation and Saving:

- Plotting accuracy and loss during training and validation allows performance to be seen.
- The trained model is saved to a file for later use in the web application.

## 2. Machine Learning Model Integration

### Model Loading and Prediction:

- The machine learning model is loaded using Keras.
- Images are pre-processed to match the model's input requirements (resized, normalized).
- Predictions are made using the loaded model, and the results are mapped to predefined categories.

### Image Processing Pipeline:

- Images are loaded and resized to the required dimensions (224x224 pixels).
- Pixel values are normalized to a [0, 1] range.
- Predictions are obtained by passing the processed image through the model.

## 3. Database Interaction (database.py)

### Database Connection

The db\_connect function establishes a connection to the MySQL database.

- connects to the database using the supplied credentials using MySQLdb.connect.
- Returns a cursor and connection object for executing queries and transactions.

### User Registration

The user\_reg function inserts new user details into the register table.

- carries out a SQL INSERT operation in order to add user data.
- Commits the transaction to save changes.
- Returns the status of the operation.

### User Login

The user\_login function validates user credentials.

- Executes an SQL SELECT statement to find matching credentials in the register table.
- If a match is found, the user session is updated.

### Image Upload

The user\_upload function inserts uploaded image details into the upload table.

- Uses session data to associate the image with the authenticated user.
- Executes an SQL INSERT statement to store image metadata.
- Commits the transaction and returns the status.

### View Images

The user\_viewimages function retrieves images uploaded by a specific user.

- Executes an SQL SELECT statement to fetch images based on the username.
- Returns the fetched data.

### Track Image Information

The v\_image function retrieves image details from the images table.

- Executes an SQL SELECT statement to fetch image data.
- Returns the fetched data for further processing.

An already-trained machine learning model is used by the image\_info function to categorise images.

- Loads the model and processes the image to make predictions.
- Returns the classification result.

## 4. Flask Web Application (app.py)

### Importing Libraries and Initializing Flask App

The Flask web application relies on various libraries for web development, database connectivity, and utility functions:

- os: Performance is visualised by plotting accuracy and loss throughout training and validation.
- MySQLdb: A Python DB-API for MySQL, used to connect and execute queries on a MySQL database.

- Flask and its submodules: The core web framework and supporting modules for session management, URL handling, and rendering templates.
- `werkzeug.utils.secure_filename`: A utility for securing filenames to avoid directory traversal attacks. The Flask application is initialized with a secret key to manage user sessions securely.

## Route Definitions

Routes in Flask define the URLs that the application will respond to and the corresponding functions to handle these requests. Each route is associated with a specific URL and a view function.

### Home and Static Pages

These routes render static HTML pages for the application's interface.

- Renders the homepage (`index.html`).
- `index.html`: Renders the homepage, often used for logout.
- `register.html`: Renders the registration page.
- `userhome.html`: Renders the user's home page after login.
- `login.html`: Renders the login page.
- `upload.html`: Renders the image upload page.
- `viewdata.html`: Renders a page to view data (images).

## User Registration

The `regact` route handles user registration.

- The registration page's form data is processed by it.
- Calls the `user_reg` function from `database.py` to insert user details into the database.
- Depending on the result, it either redirects the user to the login page or shows an error message.

## User Login

The `loginact` route manages user login.

- The login page's form data is processed by it.
- Calls the `user_loginact` function from `database.py` to validate credentials.
- If credentials are valid, it starts a user session and redirects to the user home page.
- It indicates an error if validation is unsuccessful.

## Image Upload

The `upload` route allows users to upload images.

- Processes form data and image files.
- Calls the `user_upload` function from `database.py` to store image details in the database.
- Informs the user of the success or failure of the upload.

## View Images

The `viewimage.html` route retrieves and displays images uploaded by the authenticated user.

- Queries the database for images associated with the current session's user.
- Renders the `viewimage.html` template with the retrieved data.

### Track Image Information

The `track` route handles requests for tracking specific images.

- Retrieves image name and information from the request.
- Calls `image_info` to get detailed information and `v_image` to process and return the result.
- Displays the information to the user on the `viewdata.html` template.

### 4.5 Interface Implementation

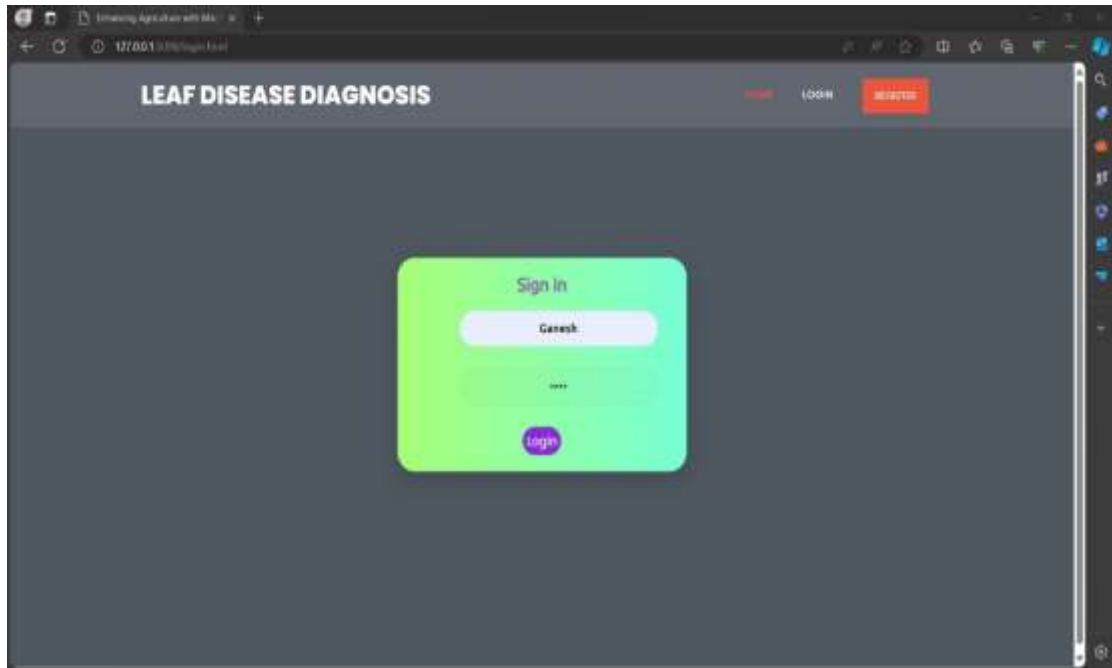


Fig 4.1: Sign in Page



Fig 4.2: Upload Image

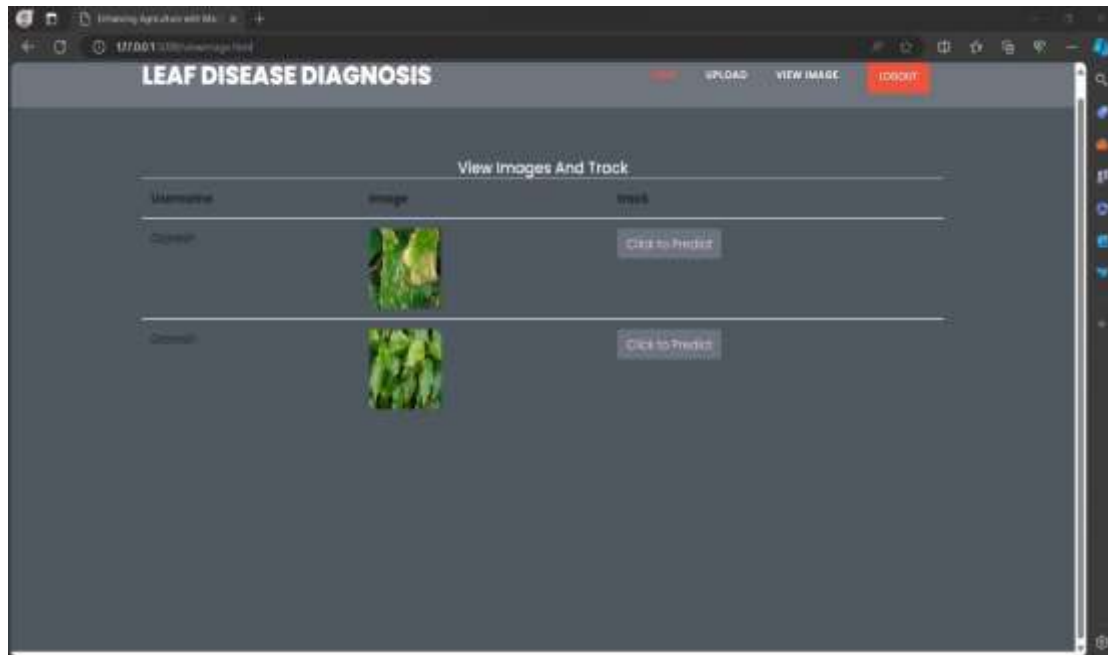


Fig 4.3: View Images and Track

### V. RESULT ANALYSIS

As we shown in *Fig 4.3: View Images and Track*, after that we need to on click predict. It will redirect to next web page as we show as below.

In that, It will test leaf and display the Disease Name and Description. In Description, it is recommends the some remedy to the user,

Based on the input image given we get the leaf disease and the remedy for the disease.

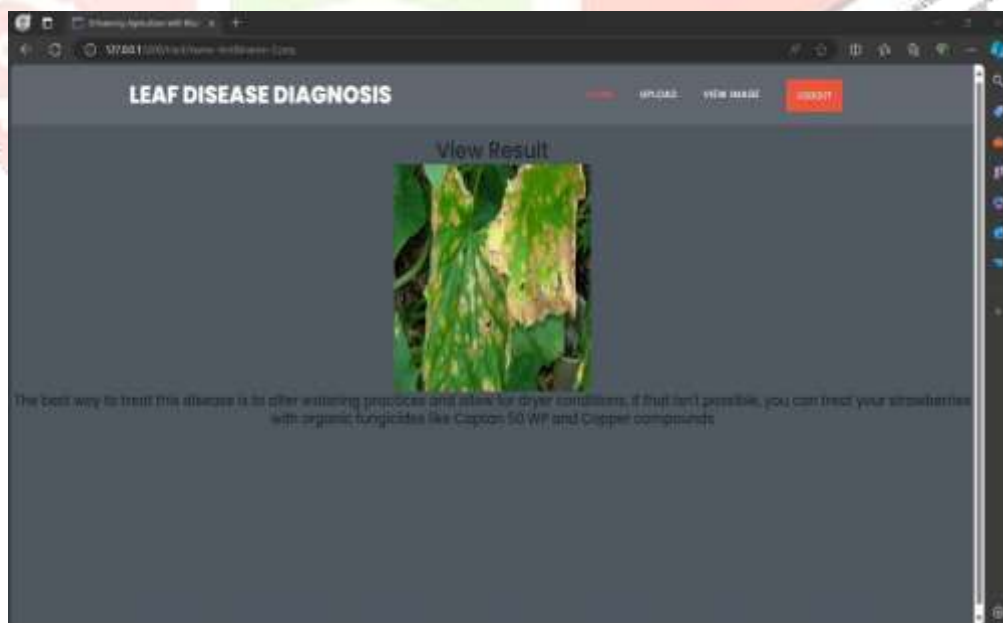


Fig 5.1: Output 1

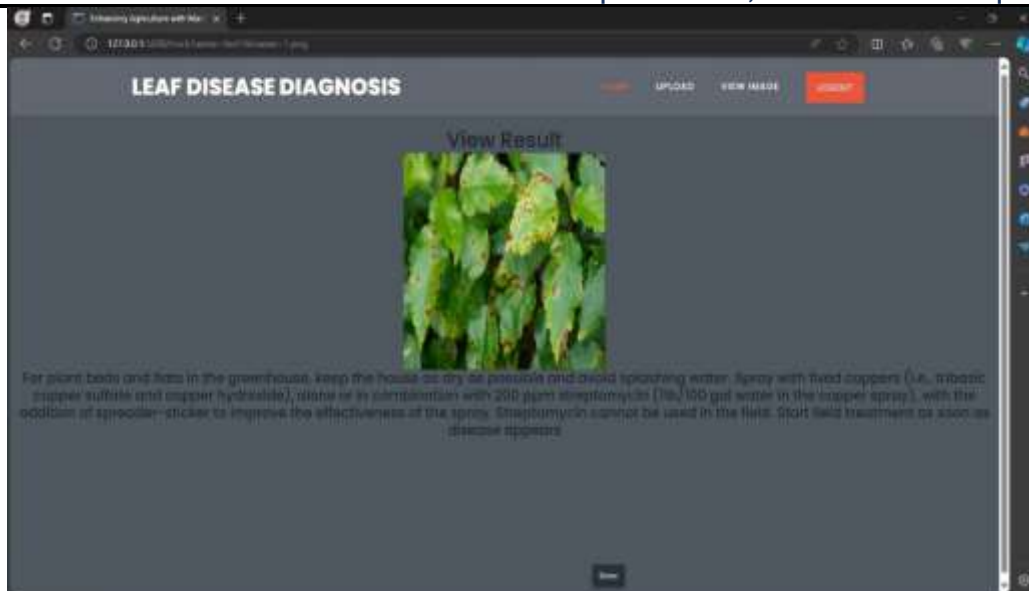


Fig 5.2: Output 2

## VI. CONCLUSION

This project offers a ready-made model that determines whether a plant leaf is sick. There are several known diseases that affect the leaves of plants, including cotton, sugarcane, wheat, and grapes. CNN Engineering handles and prepares the information. The information is prepared using MobileNet computation. For regulating the leaf infection arrangement, Tensorflow/Keras libraries are used in conjunction with Python programming.

## VII. FUTURE SCOPE

Creating a framework that satisfies every user requirement is unthinkable. The needs of the clients keep evolving while the framework is being used. Some of the future enhancements that this framework ought to enable are:

- It is possible to update the framework when new ideas emerge and adapt it to the perfect environment.
- With single sign-on and other emerging innovations, security can be further enhanced in light of future security concerns.

## REFERENCES:

- 1) P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks, "2018 Eleventh International Conference on Contemporary Computing(IC3), Noida, 2018.
- 2) P. K. Kosamkar, V. Y. Kulkarni, K. Mantri, S. Rudrawar, S. Salmpuria, and N. Gadekar, "Leaf Disease Detection and Recommendation of Pesticides Using Convolution Neural Network," 2018 Fourth International Conference on Computing Communication Control and Automation(ICCUBEA), Pune, India, 2018.
- 3) A.P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee Leaf Rust Detection Using Convolutional Neural Network," 2019 XV Workshop de Visao Computacional(WVC), São Bernardo do Campo, Brazil, 2019.
- 4) S. S. Hari, M. Sivakumar, P. Renuga, S. Karthikeyan and S. Suriya, "Detection of Plant Disease by Leaf Image Using Convolutional Neural Network, "2019 International Conference on Vision Towards Emerging Trends in Communication and Networking(ViTECoN), Vellore, India, 2019.
- 5) D. P. Sudharshan and S. Raj, "Object recognition in images using the convolutional neural network," 2018 2nd International Conference on Inventive Systems and Control(ICISC), Coimbatore, 2018.
- 6) S. Maity et al., "Fault Area Detection in Leaf Diseases Using K-Means Clustering, "2018 2nd International Conference on Trends in Electronics and Informatics(ICOEI), Tirunelveli, 2018.



- 7) M. Francis and C. Deisy, "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks — A Visual Understanding," 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2019.

