iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** I **Month of publication:** January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66511

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

Plant Disease Detection System

Deepa Kumari Giri[1], Bhumika I[2], Gagan Kumar NG[3], Bindhumeghana[4]
Dept. of Computer Science and Engineering, Ballari Institute of Technology and Management Ballari, India

Abstract: This project develops a Plant Leaf Disease Detection System that uses machine learning to identify plant diseases from uploaded leaf images. By applying a pre-trained deep learning model, the system provides real-time diagnosis and suggests remedies stored in a database. It also integrates the Google Translate API for multilingual support, making the system accessible to a global audience. The aim is to assist farmers in early disease detection, improve crop management, and promote sustainable agricultural practices.

I. INTRODUCTION

Plant disease is one of the most devastating challenges to agricultural production in a global context; hence, a quick, efficient method of diagnosis and management of plant diseases must be adopted. Manual detection of plant diseases has been very slow, subject to human error, and dependent on the experts available for interpretation. Technology-based automated systems could provide quick, correct solutions for farmers and agriculturists as a result of this challenge.

This project will build a web application for plant leaf disease detection based on machine learning, image processing, and multilingual support to provide its users reachable and practical solutions. The application is built with Flask, which serves as the backend, providing methods for prediction for disease detection and through the implementation of Google Translate API for the remedies to be available in the language of the user's choosing.

The system is designed and operated for ease of use, letting farmers upload leaf images in order to diagnose them for plant diseases, receiving predictions back immediately, along with suitable treatment options. The system provides easy diagnosis, while also being somewhat inclusive by supporting languages, making it viable for different users.

Stagewise development involved the following:

Backend implementation for authentication and image upload Implementation of an image prediction algorithm to detect disease Implementation of Google Translate API for providing multilingual remedies Database creation for disease and treatment details; and A dynamic, responsive user interface. This complete system gives real-time notifications to users about the health of their plants, working to lessen any possible damages to crop production and to allow for sustainable agriculture practices to take hold.

II. METHODOLOGY

A. *Block Diagram*

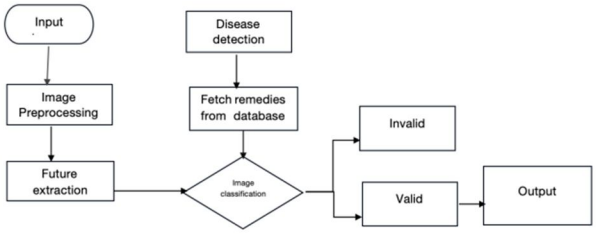


Figure 1: Plant Leaf Disease Detection System

The system follows a two-stage approach to detect diseases and provide solutions:

1) Image Upload and Preprocessing:

- In this stage, the user uploads an image of a plant leaf through the web interface. The uploaded image is then preprocessed to ensure it is in a suitable format for analysis.
- The image is cleaned (e.g., noise reduction, resizing) to improve prediction accuracy and remove unnecessary details. This helps enhance the quality of the data fed into the prediction model.

2) Disease Detection and Result Generation

- In the second stage, the preprocessed image is analyzed using the disease detection model, which leverages machine learning algorithms to predict the plant disease based on visual features.
- Once the disease is detected, the system fetches relevant remedy suggestions from the database and translates them using Google Translate API into the user's preferred language.
- The result is displayed to the user, along with possible solutions for treatment.

3) Input Image

The input image must be of a plant leaf, captured under sufficient lighting to clearly display the leaf's features. The image should be in an accepted format (e.g., JPEG, PNG) for processing by the system. Ensuring the image is clear and without significant distortion helps in achieving better accuracy during prediction.

4) Image Processing

Raw images are not directly useful for machine learning models, so preprocessing is a crucial step. The system employs several preprocessing techniques, such as:

- Image Resizing: Standardizing the image size to match the input requirements of the model.
- Feature Enhancement: Techniques such as adjusting brightness or contrast to highlight leaf features.

5) Feature Extraction

Key visual features, such as leaf shape, texture, and color, are extracted from the image using image processing techniques. These features are analyzed and passed into a machine learning model, which is trained to classify diseases based on these features. The extracted data helps in identifying the specific plant and disease.

6) Disease Detection

Using a pre-trained machine learning model, the system processes the extracted features to identify the disease affecting the plant. The model compares the visual features of the uploaded leaf image against a large dataset of known plant diseases, predicting the most likely disease. If a match is found, the system returns the disease name and possible solutions.

B. Flow Chart for Detection Process

When a user uploads an image of a plant leaf, the system begins the process of detecting the disease and providing a solution. The image is first preprocessed, then analyzed for disease prediction. If a disease is detected, the system shows relevant solutions.

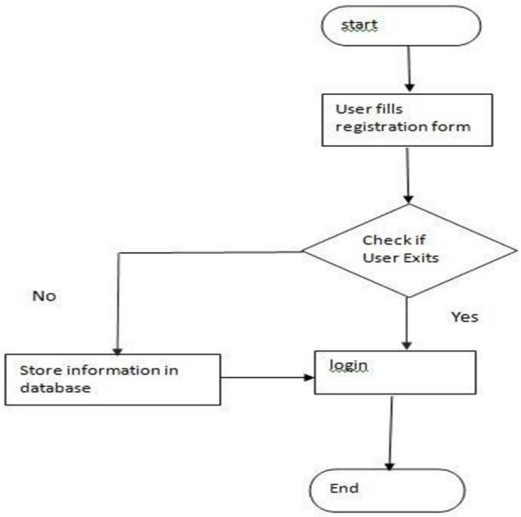


Figure 2: Flow Chart of Registration Process

C. Flow chart for Registration Process

The user registration process begins by gathering basic user details (e.g., name, email, contact number) and storing them in the database. Once a user successfully registers, they can log in to the system.

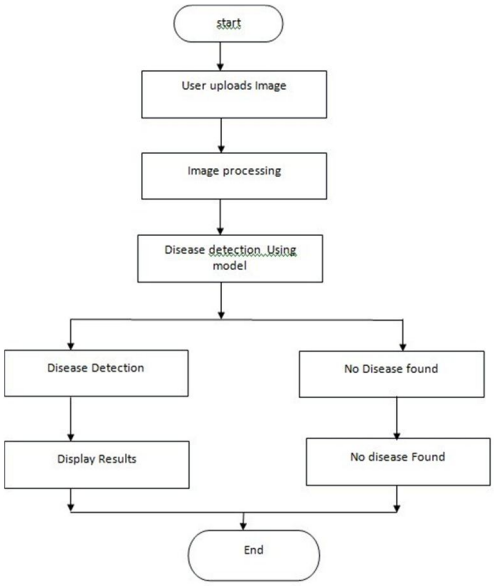


Figure 3: Flow Chart of Detection process

III. WORKING

Use case diagrams are a vital aspect of UML, serving as a visual representation of how a system interacts with its users, known as actors. These diagrams help in understanding the different scenarios where a system or application engages with individuals, organizations, or external systems to achieve specific goals.

A. Key Components

Actors: Users who interact with the system, including individuals, organizations, or external systems. They are external entities that initiate or respond to interactions with the system.

Use Cases: Represent the functionality or behavior of the system from the user's perspective. Each use case describes a particular interaction between the system and an actor, typically focusing on achieving a specific goal.

Relationships: Connections between actors and use cases, depicting how actors are involved in the system's functionality. These relationships include associations, generalizations, and extend/include relationships.

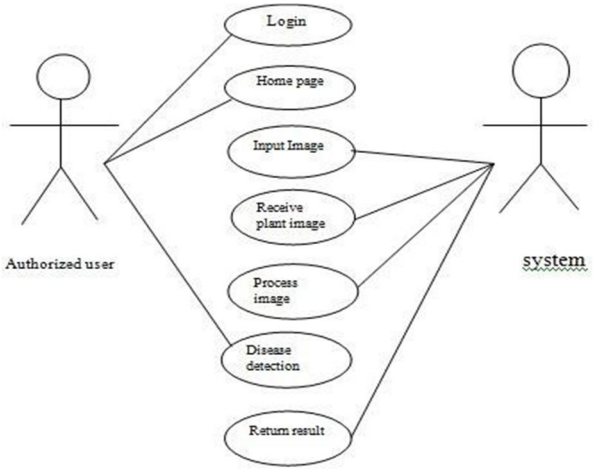


Figure 4: use case diagram of plant disease detection

B. Algorithm

VGG16 is a convolutional neural network (CNN) model used for image classification. It consists of 16 layers, including convolutional, ReLU activation, and max-pooling layers, followed by fully connected layers that classify the input.

1) Convolutional Layers: Extracts features from the input image, such as edges, textures, and patterns, that are used to identify diseases in plant leaves.
2) ReLU Activation: Applies non-linearity to the output of the convolutional layers, allowing the model to learn complex features.
3) Max Pooling: Reduces the spatial dimensions of the feature maps, making the model more efficient and reducing overfitting.
4) Fully Connected Layers: Aggregates the extracted features and outputs the final classification result.
5) Softmax Output: Provides the probability distribution across disease classes, and the class with the highest probability is selected as the prediction.

In this project, VGG16 is employed to classify plant leaves into various categories such as "Healthy" or "Diseased" by recognizing characteristic patterns in the leaves. This involves training the model on a dataset containing leaf images labeled with various disease categories, fine-tuning the model on the dataset, and then using it for real-time predictions.

IV. RESULTS AND DISCUSSION

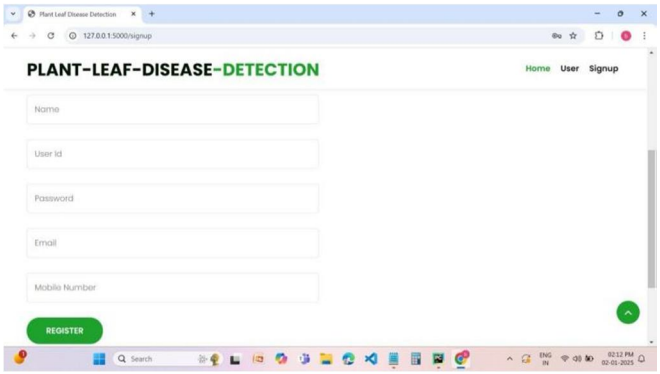


Figure 5: Registration

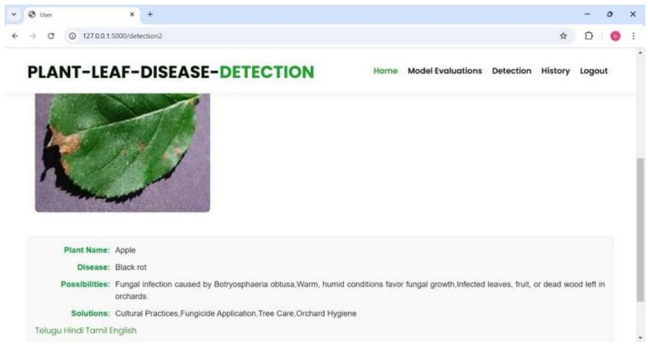


Figure 6: Detection output

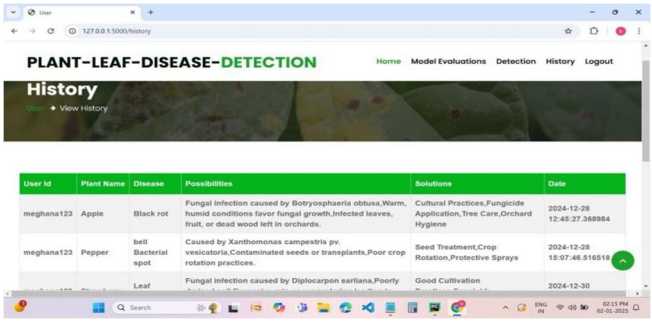


Figure 7: History

V. CONCLUSION

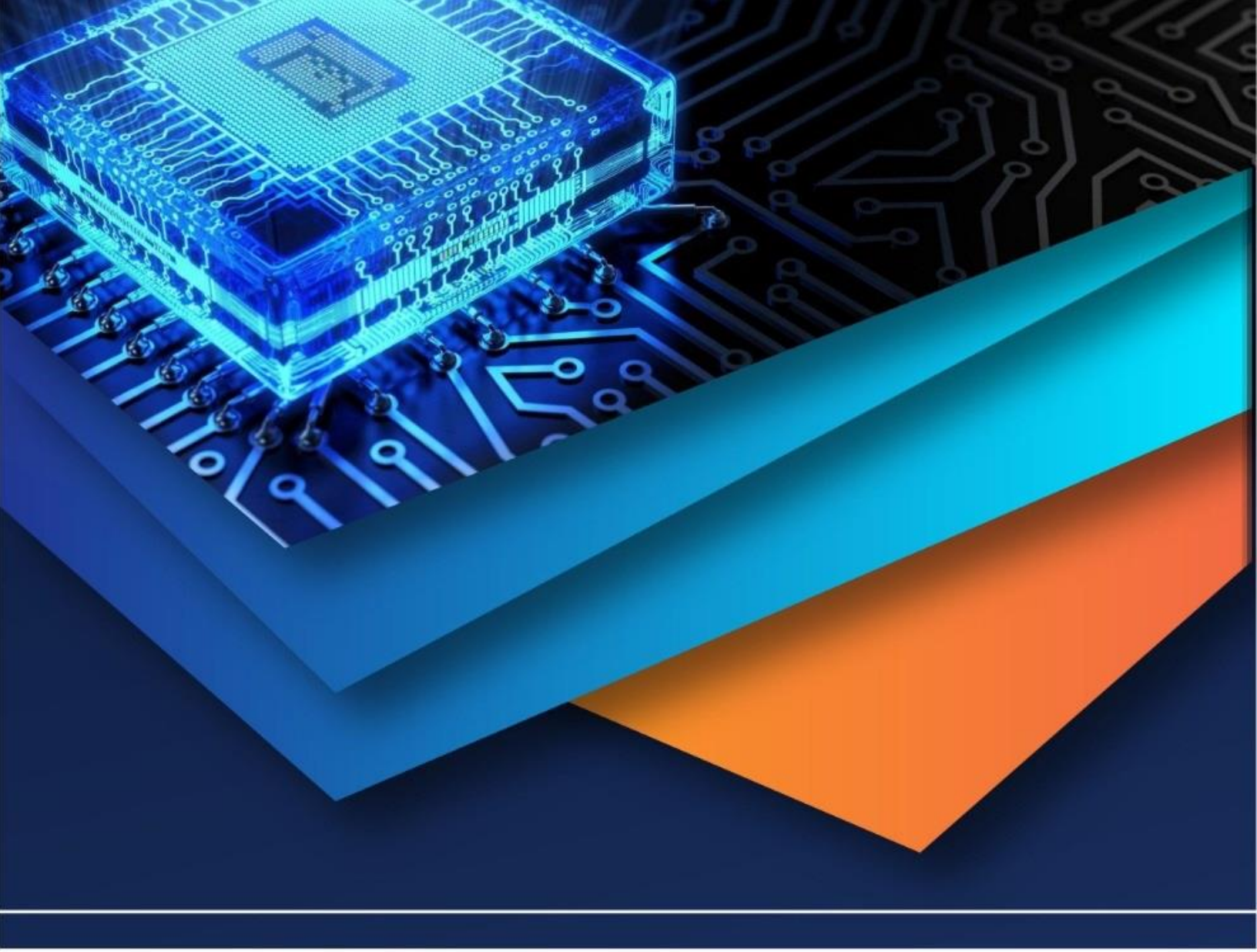
The plant leaf disease detection system using VGG16 formulates an effective and accurate mechanism for plant disease identification using image recognition. Automating the disease detection works on three notions: minimizing incorrect diagnosis, enhancing accuracy, and expediting diagnosis.

This system will assist in timely actions by farmers with respect to plant health, hence enhancing yield and quality of crops. With multilingual support, it guarantees greater accessibility.

In conclusion, this technology, if well implemented, can boost and improve agricultural activities, providing reliable and efficient resources for disease detection and management.

REFERENCES

[1] Dataset Reference: Vipooool. (2018). New Plant Diseases Dataset. Retrieved from [https://www.kaggle.com/datasets/vipoooool/new- plant-diseases-dataset].
[2] Pretrained Model (VGG16): Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from [https://arxiv.org/abs/1409.1556].
[3] Plant Disease Detection Studies: Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419.
[4] Deep Learning Framework Documentation: TensorFlow: https://www.tensorflow.org/ Keras: https://keras.io/
[5] Evaluation Metrics: Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37–63.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY