

Real Time Plant Disease Detection Using Deep Learning

Santhiya R, Velumani T

Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore,
Tamil Nadu, India
Santhiyasandy1220@gmail.com

ABSTRACT

Plant diseases cause major losses in agriculture, so early detection is important to reduce crop damage. This project uses Deep Learning, especially Convolutional Neural Networks (CNN), to detect plant diseases from leaf images. Flask is used to create a web application for real-time prediction. The system helps farmers identify diseases quickly and improve crop productivity

Keywords: Machine Learning, deep learning, cnn, Data Preprocessing, flask, leaf images, SVM, Model Evaluation

I. Introduction

Agriculture plays a vital role in the global economy and food supply, but plant diseases pose a serious threat to crop production and quality. These diseases can spread rapidly and lead to significant losses if not identified at an early stage. Traditionally, farmers rely on manual inspection to detect plant diseases, which is time-consuming, less accurate, and requires expert knowledge. With the advancement of technology, especially in the field of deep learning, automated systems can now be developed to detect plant diseases efficiently. This project focuses on designing a real-time plant disease detection system using Convolutional Neural Networks (CNN). The system processes plant leaf images, identifies disease patterns, and provides accurate classification results. Additionally, a web-based interface is developed using Flask to allow users to easily upload images and receive instant predictions. This approach not only improves detection accuracy but also helps farmers take timely actions to reduce crop damage and increase productivity.

II. Existing and Proposed System

The existing system for plant disease detection mainly depends on manual inspection and traditional machine learning methods such as SVM and KNN, which require manual feature extraction and often provide lower accuracy and slower results. In the proposed system, Deep Learning techniques, especially Convolutional Neural Networks (CNN), are used to automatically detect plant diseases from leaf images. This system improves accuracy, reduces manual effort, and provides real-time prediction through a Flask-based web application, making it more efficient and user-friendly for farmers.

III. Advantages of the Proposed System

- Provides high accuracy using CNN
- Detects plant diseases in real-time
- Reduces manual effort and human error
- Automatically extracts features (no manual work)
- Faster and more efficient than traditional methods
- Easy to use with Flask web interface
- Helps in early disease detection
- Reduces crop loss and improves productivity

III. System Specification

Hardware Requirements

- **Processor : Intel i3 / i5 or above**
- **RAM : 4 GB minimum**
- **Hard Disk : 500 GB**

- **Keyboard and Mouse**
- **Monitor**

Software Requirements

- Operating System : Windows 10 / 11
- Programming Language : Python
- Framework : Flask
- Deep Learning Library : TensorFlow / Keras
- IDE : VS Code / PyCharm / Jupyter Notebook
- Browser : Google Chrome

IV. Methodology

The methodology of this project starts with collecting plant leaf images from datasets such as PlantVillage. The images are then preprocessed by resizing, noise removal, and normalization to improve quality. After preprocessing, segmentation techniques are used to identify the infected area of the leaf. Next, feature extraction is performed automatically using a Convolutional Neural Network (CNN), which learns important patterns from the images. The extracted features are then classified into healthy or diseased categories. Finally, the result is displayed in real-time through a Flask-based web application, allowing users to upload images and get instant predictions.

V. Module Description

DATA GATHERING

Data gathering is the process of collecting plant leaf images from reliable sources such as the PlantVillage dataset. The dataset contains images of both healthy and diseased leaves from different plant species. These images are used to train and test the CNN model for accurate plant disease detection.

DATA PREPROCESSING:

Data preprocessing is the step of improving image quality before training the model. It includes resizing the leaf images, removing noise, and normalizing pixel values. These steps help the CNN model learn better features and improve disease detection accuracy.

DATA SPLITTING:

Data splitting is the process of dividing the collected dataset into training and testing sets. Usually, 80% of the images are used for training the CNN model, and 20% are used for testing. This helps evaluate the model's performance on new and unseen images.

ALGORITHM SELECTION:

Convolutional Neural Network (CNN) is selected as the main algorithm for this project because it is highly effective for image classification tasks. CNN automatically extracts important features from plant leaf images and provides high accuracy in detecting diseases. It performs better than traditional machine learning algorithms like SVM and KNN.

VI. System Design

The system is designed to detect plant diseases from leaf images using deep learning. First, the user uploads a plant leaf image through the Flask web application. The image then goes through preprocessing steps such as resizing, noise removal, and normalization. After that, segmentation is applied to identify the infected region of the leaf. A Convolutional Neural Network (CNN) extracts important features and classifies the image as healthy or diseased. Finally, the prediction result is displayed to the user in real time through the web interface.

VII. Data Flow and ER Design

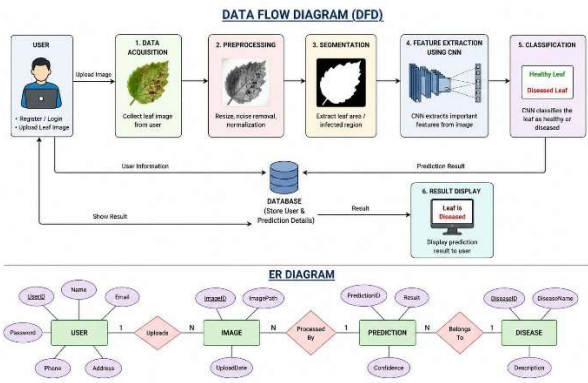


Figure 1. Data Flow Diagram

VIII. Testing and Implementation

Testing and Implementation

Testing is performed to ensure that the plant disease detection system works accurately and efficiently. The trained CNN model is tested using unseen leaf images from the testing dataset to evaluate its performance. Metrics such as accuracy, precision, recall, and F1-score are used to measure the results. After successful testing, the model is implemented using a Flask-based web application, where users can upload plant leaf images and receive real-time disease prediction results. This makes the system practical and easy to use for farmers and users.

IX. Results and Discussion

The proposed CNN-based plant disease detection system achieved high accuracy in identifying healthy and diseased plant leaves. The model performed better than traditional machine learning methods by automatically extracting important features from images. Testing results showed improved accuracy, precision, recall, and F1-score. The Flask-based web application successfully provided real-time predictions, making the system user-friendly and efficient. This approach helps in early disease detection, reduces manual effort, and supports better agricultural productivity.

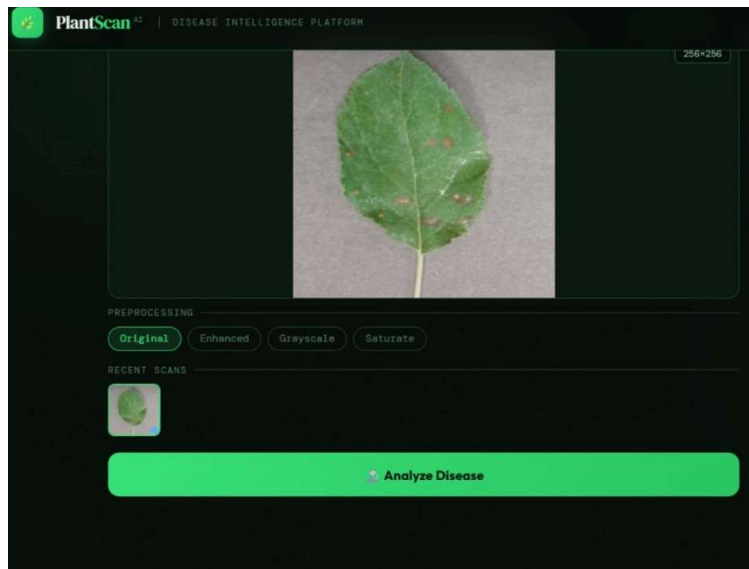


Figure 1. Abstract page from the uploaded final project report

X. Conclusion and Future Enhancement

Conclusion and Future Enhancement

The proposed plant disease detection system using Convolutional Neural Networks (CNN) successfully identifies diseases from plant leaf images with high accuracy. It provides real-time prediction through a Flask-based web application, making the system simple and useful for farmers. This approach reduces manual effort, improves early disease detection, and helps minimize crop loss.

In the future, the system can be enhanced by using larger datasets, supporting more plant species and diseases, and improving model accuracy with advanced deep learning techniques. It can also be developed as a mobile application and integrated with IoT devices for live field monitoring and smart farming solutions..

REFERENCE

- S.P. Mohanty et al., “Using Deep Learning for Image-Based Plant Disease Detection,” 2016.
K.P. Ferentinos, “Deep Learning Models for Plant Disease Detection and Diagnosis,” 2018.
S. Sladojevic et al., “Deep Neural Networks Based Recognition of Plant Diseases,” 2016.
PlantVillage Dataset.
TensorFlow Documentation.
Flask Documentation..