Project Documentation

Plant Disease Detection and Cure: A Machine Learning Driven Web Application

by
Narendra Jadhav
Akhilesh Dixit

Guide: Dr. Kirti Wanjale Industry Mentor: Sachin Deshpande

Introduction:

Plant diseases pose a significant threat to agricultural productivity and food security worldwide. The timely detection and management of these diseases are critical for minimizing crop losses and ensuring sustainable agricultural practices. Traditional methods of disease identification often rely on visual inspection by trained experts, which can be time-consuming, labor-intensive, and prone to human error. In recent years, there has been a growing interest in leveraging machine learning techniques to automate the process of plant disease detection, offering the potential for faster and more accurate diagnosis.

The aim is to present a comprehensive approach to plant disease detection using machine learning. We have developed a system that harnesses the power of convolutional neural networks (CNNs) to analyze images of plant leaves and identify the presence of various diseases. Our approach is based on a large-scale dataset obtained from Kaggle, comprising images of diseased and healthy plant specimens across multiple crop types, including apples, potatoes, corn, grapes, and tomatoes, among others.

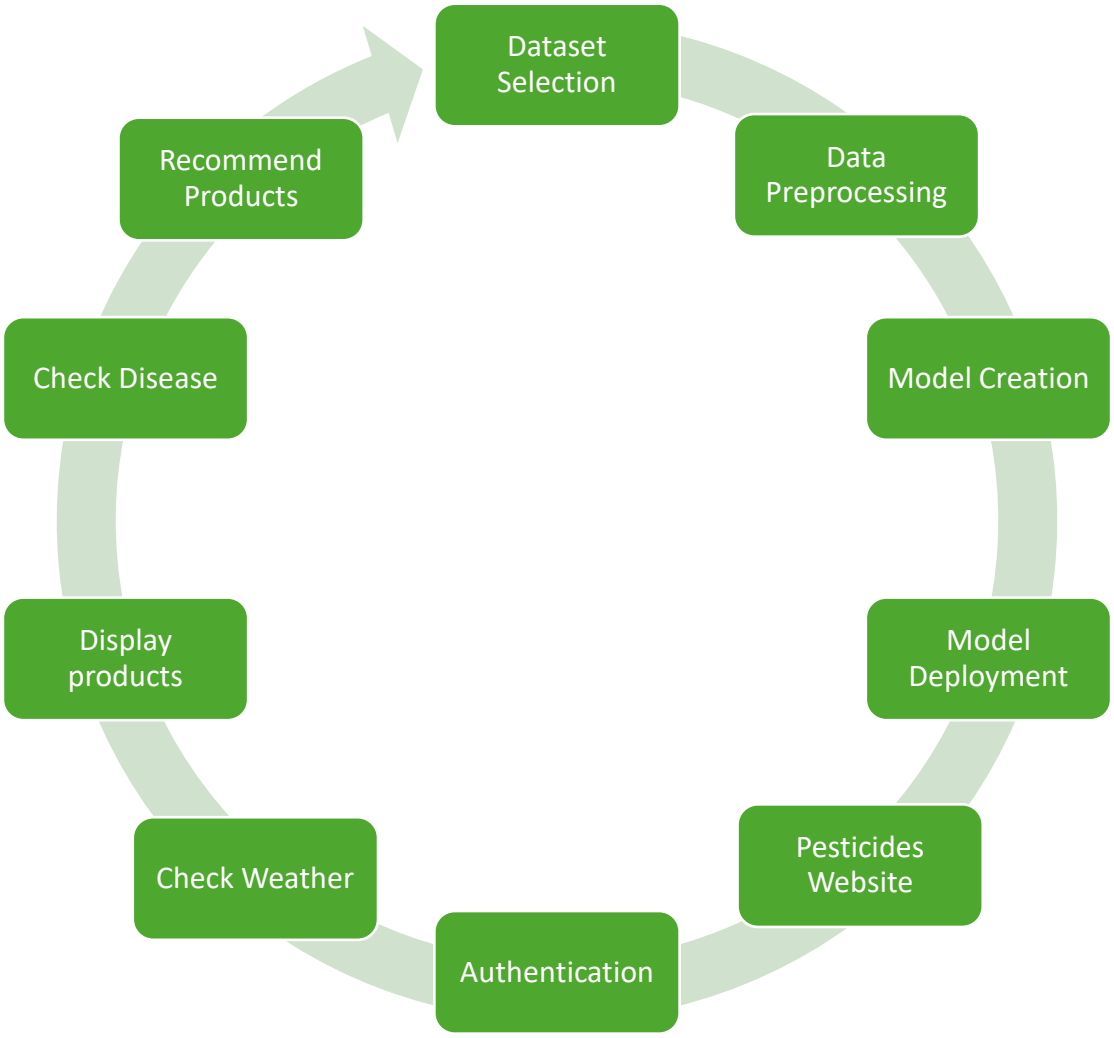
The methodology involves preprocessing the dataset to ensure uniformity and quality, including image resizing, augmentation, and normalization. A CNN model is then trained on the preprocessed images to learn patterns and features indicative of different plant diseases. The trained model is capable of accurately classifying images into one of several disease categories, enabling farmers and agricultural experts to quickly identify and respond to potential threats in their crops.

In addition to the development of the machine learning model, our research also encompasses the integration of this model into a user-friendly web application. Built using React.js, Node.js, MongoDB, and Tailwind CSS, the application provides a seamless interface for users to upload images of diseased plants and receive instant disease predictions. Furthermore, the application includes features such as user authentication, a weather forecasting section, and product recommendations based on the detected disease.

Through this research, we aim to demonstrate the potential of machine learning in revolutionizing plant disease management and agricultural practices. By combining advanced algorithms

with modern web technologies, we offer a holistic solution that empowers farmers and stakeholders with the tools they need to combat plant diseases effectively and sustainably.

Overview:



Dataset Used:

https://www.kaggle.com/datasets/dixitakhilesh/plant-disease-dataset

Created by combining the following datasets:

https://www.kaggle.com/datasets/emmarex/plantdisease

https://www.kaggle.com/datasets/lavaman151/plantifydr-dataset

Dataset Description:

The dataset used in this research comprises images of plant leaves affected by various diseases, as well as images of healthy plant specimens. The dataset was obtained from Kaggle, a platform for sharing and discovering datasets, and it covers a diverse range of crop types, including apples, potatoes, corn, grapes, tomatoes, and wheat, among others.

The dataset contains a total of 35 classes, with each class representing a specific plant disease or the healthy state of a particular crop. Each class includes a varying number of images, providing a comprehensive representation of the different disease manifestations across multiple plant species.

The bar chart below (Fig 1) illustrates the distribution of images across various plant disease classes in the dataset. Each bar represents a different plant disease class, and the height of the bar indicates the number of images available for that class. The x-axis displays the names of the plant disease classes, while the y-axis represents the corresponding number of images. The chart provides a quick overview of the distribution of images among different plant diseases, aiding in understanding the dataset's composition and balance.

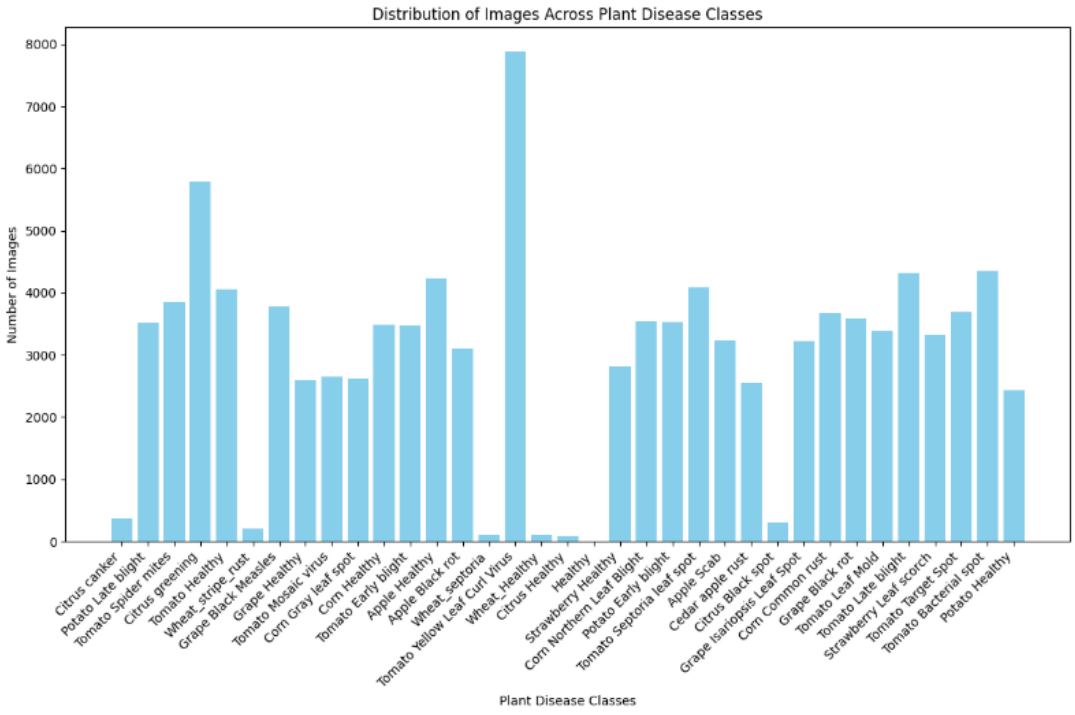


Fig 1: Distribution of images across Plant Disease classes

Plant Diseases Cure:

- https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/citrus-black-spot
- https://extension.umn.edu/plant-diseases/black-rot-apple#:~:text=Captan%20and%20sulfur%20products%20are,as%20the%20infection%20of%20fruit.

- https://www.mdpi.com/2223-7747/10/12/2643
- https://extension.wvu.edu/lawn-gardening-pests/plant-disease/fruit-vegetable-diseases/late-blight-tomatoes#:~:text=For%20conventional%20gardeners%20and%20commercial,to%20prevent%20late%20blight%20infection.
- https://www.cropscience.bayer.us/articles/bayer/managing-northern-corn-leaf-blight

Tech Stack used:

- Frontend Framework: React.js
- Backend Framework: Node.js
- Database Management: MongoDB
- Styling Framework: Tailwind CSS
- Authentication: Clerk - https://clerk.com/
- API Integration: OpenWeatherMap API, GeoDB Cities API
- Machine Learning: CNN
- Model Deployment: Flask, Hugging Face
- Website Deployment: Vercel

Code:

1. Model Development:

https://colab.research.google.com/drive/1RIA6M8_IOcycwatoj8mJSfrpBYmUQDgr

This ipynb file requires the dataset and using that the model is created which is further deployed

2. Model Deployment:

https://github.com/DixitAkhilesh/plant-disease-detection

The model is deployed on the Hugging face website - https://huggingface.co/spaces/dixitakhilesh/plant-disease-detection

This can be used to include in the main website in the 'Check-Disease' section

3. Website:

https://github.com/Narendra-Jadhav/green-shield

This contains backend - Nodejs and frontend – Reactjs

    i. Backend:
        a. npm install

    b. npm start
ii. Frontend:
    a. npm install
    b. npm run dev
iii. Backend .env file:
    a. PORT
    b. MONGO_URI
iv. Frontend .env file:
    a. VITE_CLERK_PUBLISHABLE_KEY
    b. VITE_RAPID_API_KEY
    c. VITE_RAPID_API_HOST
    d. VITE_WEATHER_API_URL
    e. VITE_WEATHER_API_KEY
    f. VITE_GEO_API_URL
    g. VITE_BACKEND_DEPLOYMENT_URL

Project Demonstration Video:

https://drive.google.com/file/d/1cVrBMAO3-7gDey_3u6QLJ0QcgmPa6TtF/view?usp=sharing