CROP-D

Project Report

2022-11-28

Made By:

1) Shoray Singhal (LCI2020037)
2) Pranav Gade (LCI2020010)
3) Vishnu Nithin Reddy (LCI2020018)

Submitted to:

Dr. Naveen Saini

Introduction

This project aims to detect and classify crop diseases so that farmers can use our tool to help them detect and respond to disease outbreaks. We use Machine Learning to select the extracted features, train a model on the features and deploy the model in a flask backend to allow API access.

TITLE OF THE PROJECT

“Classification and Recognition of Plant Diseases ”

Our **CROP-D system** a utility to help farmers detect and classify plant diseases using their leaves. It consists of an intuitive web frontend and has an integrated detection and classification model for Plant disease detection. Initially, it have support for common leafy crops like Peppers, Tomatoes, Potatoes, etc. The user has to upload the image of the ailing plant and the models will predict what plant it is and return the disease with probable cures suggestions for the same.

Why such a System?

A large number of crops and harvests are lost to plant diseases every year and farmers have no reliable way to identify and cure these on time. Also they have no reliable way to find cures for the same.

Main Goal of The Project

The main goal of the project is to help farmer tackle all the above problems from a single platform.

The process is done as:

- The user uploads an image of the plant leaf that is suspected to have a disease.
- The frontend sends a request to our backend about the image.
- Our trained model identifies the disease and then returns it to the user along with probable cures.

Advantages of the our System

- Economical than traditional methods
- Instant results
- Thin-Client Architecture
- Accuracy of 95.4%
- Cross-Platform

Why we use SvelteKit

The frontend for the app is made in SvelteKit which is a up and coming web framework for making quick and low latency webapps with support for thin-client systems.

It also provide dynamic and automatic routing and can easily be deployed on the web a simple build step and also has excellent support for dynamic rendering and backend integration for servers on the edge.

Why we use sklearn

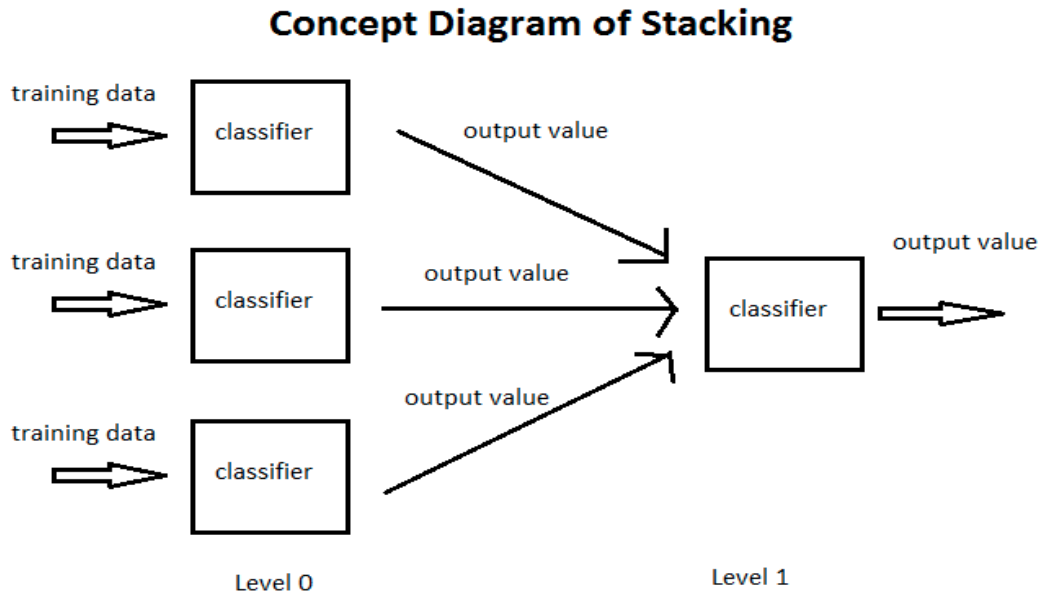
The Machine Learning part of the project is done predominantly in sklearn as it contains a lot of easy to use and efficient implementations of popular machine learning algorithms so we didn’t had to build them from scratch.

Why we use Flask

The Backend of the project was implemented in Flask which is a famous python based framework for making APIs with quick response times.

About the Model

The model of the project works on a concept of Ensemble Learning called “Stacking”.



This uses features obtained after feature extraction from the given image and then Feature Selection to select the most relevant features. The model uses a variety of classifiers such as Random Forest, KNN, Classifier Tree etc to apply stacking to. Using this we were able to achieve an accuracy of 95.4 during testing.

Relevant Links

- Deployment:
  - https://crop-d.netlify.app/
- GitHub:
  - https://github.com/Shoray2002/CROP-D
- Presentation:
  - Canva