Individual Report

Crop Disease Detection and Prediction Project

Member: Cassiel L.

Roll No.: 24P0051006

Institute: Goa University

Introduction

The **Crop Disease Detection and Prediction** project aims to integrate weather-based data and image recognition to detect crop diseases early. This multi-phase system helps farmers monitor crop health and predict disease risk based on both *environmental* and *visual* cues. My contribution focused on the **Weather API Integration and Predictive Model Training** module.

Objective

The key objective of my component was to:

- Collect real-time weather data using the **OpenWeatherMap API**.
- Process and label this data for machine learning.
- Train a **Random Forest Classifier** to predict disease-conducive conditions.
- Save and manage data within Google Drive and GitHub.

Tools and Technologies Used

- **Language:** Python
- **Libraries:** pandas, requests, sklearn, joblib
- **Platform:** Google Colab linked with GitHub
- **API:** OpenWeatherMap (Real-time weather data)

Methodology

Data Collection

Weather data was fetched for six Indian cities: *Panaji, Mumbai, Delhi, Chennai, Kolkata, and Pune*. The OpenWeatherMap API provided live data on temperature, humidity, rainfall, pressure, and wind speed.

A sample request looked like:

```
url = f"https://api.openweathermap.org/data/2.5/weather?q={city},I
N&appid={API_KEY}&units=metric"
r = requests.get(url).json()
```

The conditions were then labeled:

If humidity $> 80\%$ and $20°C \leq$ temperature $\leq 35°C$ and rainfall $> 2mm \Rightarrow$ Label as **High Risk (1)**, else **Low Risk (0)**.

Model Training

The collected dataset was stored as:

`/content/drive/MyDrive/crop_project/data/weather_dataset.csv`

A Random Forest model was trained using:

```
X = df[["temp", "humidity", "rainfall", "pressure", "wind_speed"]]
y = df["label"]

model = RandomForestClassifier()
model.fit(X, y)
```

The trained model was then saved as:

`/content/drive/MyDrive/crop_project/models/weather_model.pkl`

Integration

The weather model forms the analytical foundation for environmental risk detection. Its predictions can be combined with the CNN-based image analysis module to form a hybrid prediction system.

Results

- Successfully collected real-time weather data for multiple locations.
- Generated and saved dataset with seven key parameters.
- Trained and exported a reliable Random Forest model.
- Pushed model and dataset to GitHub under the `Cassiel` branch.

Challenges Faced

- Handling API rate limits and missing data responses.
- Managing file paths between Colab, Drive, and GitHub.
- Ensuring consistent data formats during live collection.

Conclusion

The weather-based prediction model successfully identifies favorable conditions for crop diseases. When combined with the CNN visual detection model, the project offers a powerful and holistic system for smart agriculture.

Individual Learning Outcomes

- Learned how to use and authenticate public APIs.
- Understood real-time data processing in machine learning workflows.
- Enhanced version control skills using Git and GitHub.
- Improved understanding of environmental influence on crop health.

— End of Report —