Plant Disease Detection System

This project is a Deep Learning application capable of detecting plant diseases from leaf images. It uses a Convolutional Neural Network (CNN) trained on the PlantVillage dataset and provides a local GUI for easy prediction.

📋 Table of Contents

📂 Project Structure

Ensure your local project folder is organized exactly like this to avoid "File Not Found" errors:

```
/Your_Project_Folder
│
├── gui_detector.py             # Main application script (GUI)
├── analysis_logic.py           # Helper script for prediction logic
├── Plant_Disease_Model_Final.h5 # The trained Model (Large file ~75MB)
├── class_names.json            # List of disease names (e.g., "Tomato_Healthy")
└── README.md                   # This documentation file
```

☁️ Phase 1: Training in the Cloud

We use **Google Colab** for training because it provides free GPUs. Training locally on a laptop CPU is too slow.

1. Prepare the Data (The "Zip" Trick)

Do not upload the extracted dataset folder directly to Colab. It will crash the browser attempting to upload 50,000 small files.

1. Download the **PlantVillage** dataset.
2. Right-click the dataset folder and **Compress to ZIP** (name it `plant_dataset.zip`).
3. Upload `plant_dataset.zip` to your **Google Drive** root folder.

2. Configure Colab

1. Go to colab.research.google.com.

2. Create a new notebook.
3. **Enable GPU:** Go to `Runtime` > `Change runtime type` > Select **T4 GPU**.
4. Mount Drive and unzip the data in the notebook:

    ```
    from google.colab import drive
    drive.mount('/content/drive')
    !unzip "/content/drive/My Drive/plant_dataset.zip" -d "/content/dataset"
    ```

💻 Phase 2: Local Setup

Once training is complete, download the `.h5` model file and the `.json` class names to your laptop.

1. Install Dependencies

Open your terminal (or VS Code terminal) and install the required libraries:

```
pip install tensorflow numpy pillow
```

Note: If you have an NVIDIA GPU on your laptop, ensure you install the CUDA-compatible version of TensorFlow, though the CPU version works fine for simple predictions.

🐙 Phase 3: Version Control (GitHub)

Because the model file (`.h5`) is roughly **75MB**, you cannot use the GitHub website's "Drag and Drop" feature (limit 25MB). You must use VS Code or Git Terminal.

1. Initialize Git

Open your project folder in **VS Code**, open the terminal (`Ctrl +`), and run:

```
git init
git add .
git commit -m "Initial commit with model"
```

2. Push to GitHub

1. Create a **new empty repository** on GitHub (do not add a README/gitignore yet).
2. Link your local folder to the remote repo:

    ```
    git remote add origin [https://github.com/YOUR_USERNAME/YOUR_REPO_NAME.git](https:/
    git branch -M main
    ```

```
git push -u origin main
```

⚠️ Important Note on File Size

- **Github Website Limit:** 25 MB
- **Git Push Limit:** 100 MB
- **Your Model:** ~75 MB

Status: This will work via VS Code/Terminal. If your model grows **larger than 100MB** in the future, you must use **Git LFS**:

```
git lfs install
git lfs track "*.h5"
```

🚀 Usage

To run the detector on your laptop:

1. Open the terminal in your project folder.
2. Run the main script:

```
python gui_detector.py
```

3. A window will appear. Click **"Upload Image"**, select a leaf image, and the model will display the disease prediction and confidence score.

Troubleshooting

- **Error:** `No such file or directory: 'model.h5'`
  - Make sure your terminal is actually inside `/Your_Project_Folder`. Type `ls` (Mac/Linux) or `dir` (Windows) to verify the file exists.
- **Error:** `OSError: SavedModel file does not exist`