



GREENAID README

 Folder/File	 Description
<code>01_Neural_Network.ipynb</code>	Standalone Neural Network for general classification.
<code>02_cnn_plant_type.ipynb</code>	Model for detecting plant type (Tomato, Potato, Bell Pepper).
<code>03_cnn_disease_type.ipynb</code>	Model for classifying disease status (Healthy / Infected).
<code>04_Bounding_Boxes.ipynb</code>	YOLO-based model for localizing disease spots.
<code>05_Final_Model.ipynb</code>	Combines Plant, Disease, Bounding Box + LLM for treatment advice.
<code>data/</code>	Folder with 1) data for yolo bounding boxes 2) data for all other models.

How to Run

Below is a simple **step-by-step flow** to run the entire project.

1 Train Standalone Neural Network

Step	Command
Open	<code>01_Neural_Network.ipynb</code>
Run	All cells
Save	Save for further use

Dataset -  `archive/PlantVillage/`

✓2 Train Plant Detection Model

Step	Command
Open	<code>02_cnn_plant_type.ipynb</code>
Run	All cells
Save	Save and Export best model for <code>final model notebook</code>

Dataset -  `archive/PlantVillage/`

✓3 Train Disease Detection Model

Step	Command
Open	<code>03_cnn_disease_type.ipynb</code>
Run	All cells
Save	Save and Export best model for <code>final model notebook</code>

Dataset -  `archive/PlantVillage/`

✓4 Train Bounding Box Model

Step	Command
Open	<code>04_Bounding_Boxes.ipynb</code>
Run	All cells
Save	Save and Export best model for <code>final model notebook</code>

Dataset -  `YOLO/ (custom annotated images)`

✓ 5 Combine All — Final Pipeline

Step	Command
📌 Open	05_Final_Model.ipynb
▶ Load	Insert All best models from each of the notebooks in required lines of the code
⚙️ Run	All cells — this will:
	✓ Detect Plant Type
	✓ Classify Disease
	✓ Draw Bounding Box
	✓ Pass to LLM for treatment recommendation

🔧 Dependencies

Before running, make sure you install:

#CODE

```
pip install tensorflow keras numpy opencv-python pillow matplotlib  
tqdm yolov5 transformers
```

Key Points

✓ **data organization:**
/data/PlantType/DiseaseClass/images

✓ **consistent image size:**
Set `img_size` the same in all notebooks.

✓ **Export best models:**
Save as `.h5` for Keras or `.pt` for YOLO.

✓ **Load paths correctly:**
The final notebook must load the correct saved weights.