PROJECT PLANT DISEASE CLASSIFICATION

PROJECT TIER 1

RODRIGO SIERRA

1378 COMPUTER VISION

TEACHER: PATRICIA MCMANUS

DECEMBER 04 2025

Image source: https://gardenerspath.com/how-to/disease-and-pests/septoria-leaf-spot-tomatoes/

Image source: https://www.iowapublicradio.org/gardening/tomato-diseases-in-iowa/

Image source: https://www.starnewsonline.com/story/news/2021/03/10/burgaws-thomas-produce-farm-famous-tomatoes-close/6946296002/

WHAT IS THE PROBLEM?

What real-world problem are you solving?

Who cares about this problem?

Why is it important?

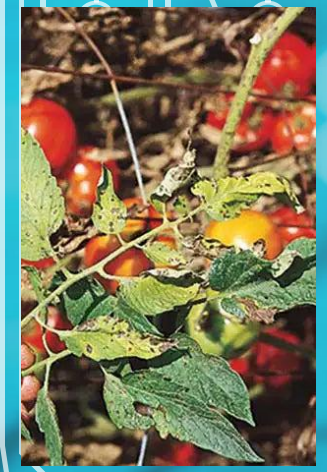


Image source: https://www.gurneys.com/pages/tomato-diseases-and-treatment?srsltid=AfmBOoqaZuStZXv5Myhn9-spD1pGL0tw3brCynRtcGi13ivfG3uJ7mp2//

Project Structure for Plant Disease Classification

1. Environment and Data Preparation
2. Data Preprocessing
3. Model and Training
4. Training and Evaluation (Training Loop)
5. Final Evaluation (Metrics and Prediction)

```
--- Prediction ---
Predicted Class: Orange___Haunglongbing_(Citrus_greening)
Confidence Score: 100.00%
```

Predicted: Orange___Haunglongbing_(Citrus_greening)
Confidence: 100.00%

```
--- Demonstration Result ---
Image classified as: Orange___Haunglongbing_(Citrus_greening)
Confidence: 100.00%
```

MY SOLUTION

What will my system do?

How will it solve the problem?

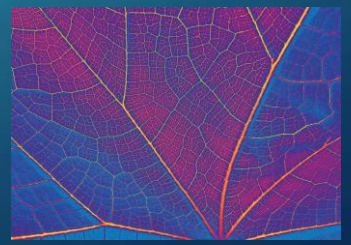
TECHNICAL APPROACH?

- **CV Technique: Classification:**

This project uses image classification to identify plant leaf diseases across 38 categories. **Model: ResNet50.**

- **Framework: PyTorch:**

We use a pre-trained ResNet50 model because it already knows how to recognize important visual patterns like colors, textures, and spots on leaves.



TECHNICAL APPROACH?

- **Framework: PyTorch:**

PyTorch is used because it is easy to work with, handles images efficiently, and provides pre-trained models like ResNet50.

- **Why this approach?**

High accuracy with less training.

Strong feature extraction from ResNet50

Works well with different plants

DATASET PREPROCESSING

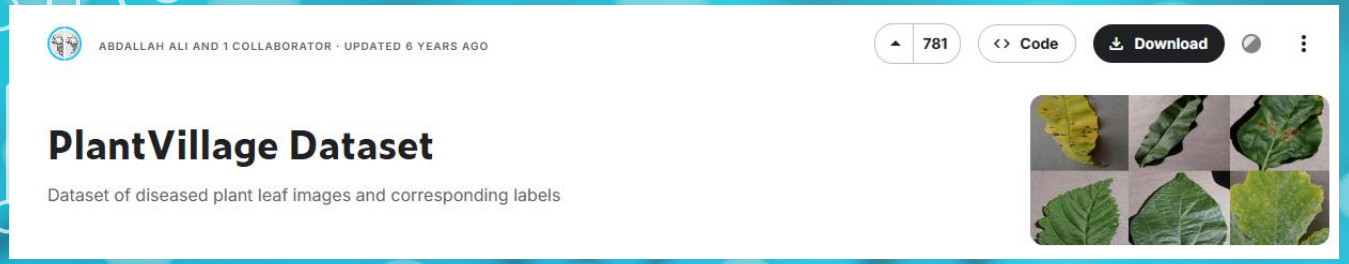
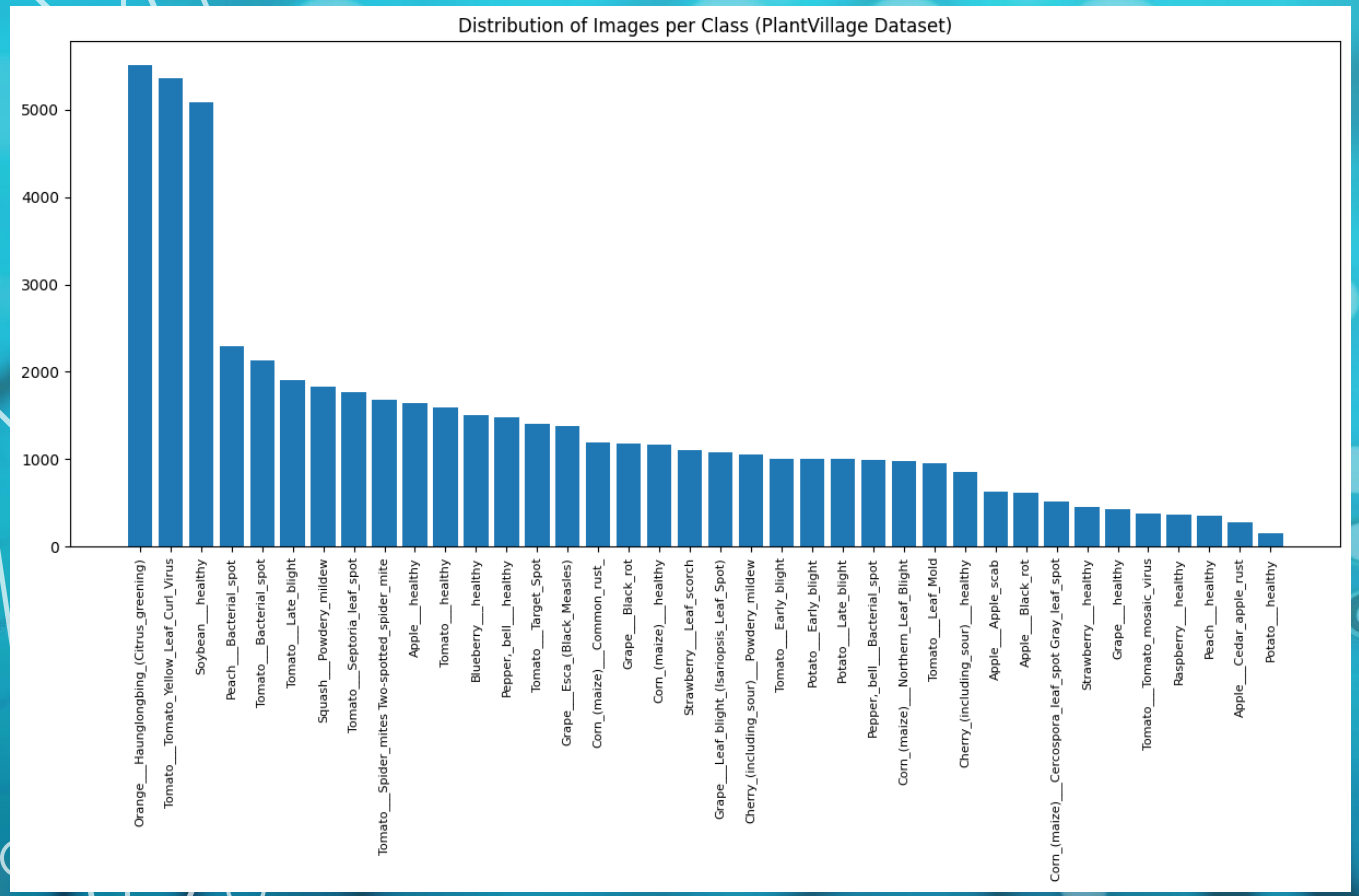


Image source: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

- **Source: PlanVillage Dataset.**
- **Size: 54.000+ images.**
- **Labels: 38 different classes.**
- **Link: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset**

My dataset contains images grouped by class in separate folders. Each class represents a specific condition the model needs to recognize. Preprocessing makes them consistent before training. I used this folder structure directly with PyTorch's ImageFolder to load the data.



DATASET PREPROCESSING

- Dataset not balanced
- This difference matters because it can affect how well the model learns each class.
- Data augmentation

DATASET PREPROCESSING

```python
# Image size required by ResNet50
IMG_SIZE = 224
IMAGENET_MEAN = [0.485, 0.456, 0.406]
IMAGENET_STD = [0.229, 0.224, 0.225]
# Batch size for load images (training)
BATCH_SIZE = 32
# Transformations for training (Augmentation and Normalization)
train_transforms = transforms.Compose([
    transforms.RandomResizedCrop(IMG_SIZE),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize(IMAGENET_MEAN, IMAGENET_STD)
])
# Transformations for validation and test (Resizing and Normalization)
val_test_transforms = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(IMG_SIZE),
    transforms.ToTensor(),
    transforms.Normalize(IMAGENET_MEAN, IMAGENET_STD)
])
```

I organized the images into folders by class, resized them so they all have the same shape, and normalized them so the model can learn better. I also added basic augmentations like flips and rotations to help the model handle variations. Then I loaded everything using PyTorch's ImageFolder and DataLoader.

- **Source: PlanVillage Dataset.**
- **Size: 54.000+ images.**
- **Labels: 38 different classes.**
- **Link: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset**

LIVE DEMO

RESULTS

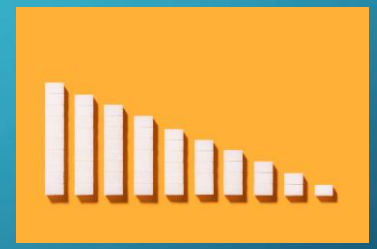
```
Epoch 8/10
Validation Acc: 0.9507
Epoch 9/10
Validation Acc: 0.9558
Epoch 10/10
Validation Acc: 0.9494

Best Validation Accuracy: 0.9580
```

RESULTS

--- Classification Report ---

	precision	recall	f1-score	support
Apple___Apple_scab	0.95	0.87	0.91	63
Apple___Black_rot	1.00	0.98	0.99	62
Apple___Cedar_apple_rust	1.00	0.93	0.96	27
Apple___healthy	0.95	0.96	0.96	165
Blueberry___healthy	0.99	0.99	0.99	150
Cherry_(including_sour)___Powdery_mildew	0.98	0.99	0.99	105
Cherry_(including_sour)___healthy	0.99	1.00	0.99	85
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0.75	0.90	0.82	51
Corn_(maize)___Common_rust_	0.99	1.00	1.00	119
Corn_(maize)___Northern_Leaf_Blight	0.94	0.85	0.89	99
Corn_(maize)___healthy	1.00	1.00	1.00	116
Grape___Black_rot	0.99	0.95	0.97	118
Grape___Esca_(Black_Measles)	0.96	0.99	0.97	138
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	1.00	1.00	108
Grape___healthy	1.00	0.98	0.99	42
Orange___Haunglongbing_(Citrus_greening)	1.00	1.00	1.00	551
Peach___Bacterial_spot	0.97	1.00	0.99	230
Peach___healthy	0.84	1.00	0.91	36
Pepper,_bell___Bacterial_spot	0.96	0.99	0.98	100
Pepper,_bell___healthy	0.97	0.98	0.98	148
Potato___Early_blight	0.92	0.94	0.93	100
Potato___Late_blight	1.00	0.71	0.83	100
Potato___healthy	0.93	0.93	0.93	15
Raspberry___healthy	0.97	1.00	0.99	37
Soybean___healthy	0.99	0.99	0.99	509
Squash___Powdery_mildew	1.00	0.99	1.00	184
Strawberry___Leaf_scorch	0.99	0.98	0.99	111
Strawberry___healthy	0.94	0.98	0.96	46
Tomato___Bacterial_spot	0.96	0.91	0.93	213
Tomato___Early_blight	0.91	0.73	0.81	100
Tomato___Late_blight	0.83	0.95	0.89	191
Tomato___Leaf_Mold	0.96	0.91	0.93	95
Tomato___Septoria_leaf_spot	0.92	0.92	0.92	177
Tomato___Spider_mites Two-spotted_spider_mite	0.91	0.89	0.90	168
Tomato___Target_Spot	0.82	0.88	0.85	140
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.98	0.98	0.98	536
Tomato___Tomato_mosaic_virus	0.66	1.00	0.80	37
Tomato___healthy	0.98	0.99	0.98	159
accuracy			0.96	5431
macro avg	0.95	0.95	0.94	5431
weighted avg	0.96	0.96	0.96	5431



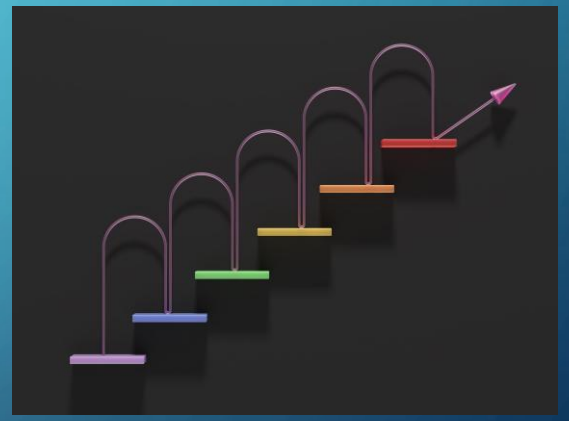
WHAT WORKED / CHALLENGES OVERCOME

I overcame a lot of issues:

- Kaggle
- Mount Google Colab
- Store Google Drive
- Demo after training model
- And many lines in the code

I have learned that:

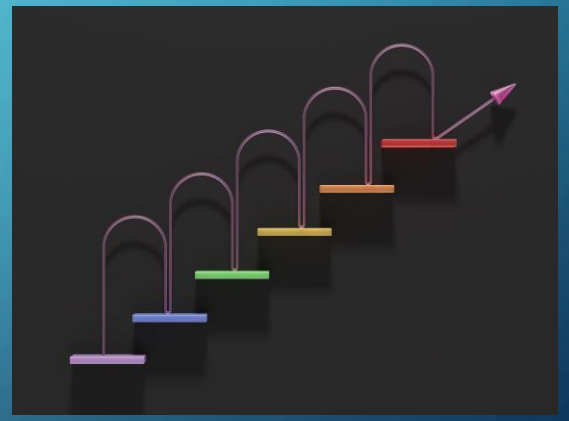
- All of these issues allow me to realize the importance to learn more about Python, and interact with different resources for future projects.
- I need to do more exercises (projects) to improve my skills related with the use of these tools.



FUTURE WORK

After this class, I need to practice with other type of tiers project to understand more complex knowledge.

May be, try to integrate the theory and practical perspective of this class with other class in the program. For example, robotics.



ACKNOWLEDGMENTS

Thank you professor for all of your dedication, explanation, and tools that helped me to understand and developed the project.

Thank you teammates because every intervention in class, added value to my knowledge and help me to think in other possibilities to do things.

Thank you AI tools like Chatgpt that help me to understand concepts required for this project.