

Plant Disease Detection System

ASSIGNMENT SUBMISSION | PRE-ASSESSMENT TASK

| Project Overview

🎯 The Objective

To assist farmers in early plant disease detection using an AI-powered computer vision solution. The goal is to provide a lightweight, accurate tool integrated into the Syngenta Mobile App ecosystem.

📦 The Tech Stack

Model: EfficientNetV2B0 (Transfer Learning)

Framework: TensorFlow / Keras

Interface: Gradio Web UI

Deployment: Mobile-Optimized (~24MB)

The Challenge

Visual identification of plant diseases is error-prone and requires expert knowledge. Delayed diagnosis leads to crop loss.

Our Solution: Automated Real-time Classification

Model Architecture: EfficientNetV2B0



Transfer Learning

Fine-tuned on custom dataset for 10-class classification, leveraging pre-trained ImageNet weights.



Lightweight Design

Approximately **24MB** in size. optimized for mobile and edge deployment where bandwidth is low.



High Performance

Achieves **~98%** validation accuracy with faster training speeds and lower parameter count.

Target Crops & Classes

CROP TYPE	DISEASE CLASSES	STATUS
Apple	Scab, Black Rot, Cedar Rust, Healthy	✔ Active
Cherry	Powdery Mildew, Healthy	✔ Active
Corn	Gray Leaf Spot, Common Rust, Northern Leaf Blight	✔ Active
Blueberry	Healthy (Baseline Class)	✔ Active

| Performance Metrics

98%

ACCURACY

10

CLASSES

4

CROP TYPES

Evaluation Strategy

Confusion Matrix Analysis

We utilize confusion matrices to identify specific misclassifications between visually similar diseases (e.g., Apple Scab vs. Black Rot), allowing for targeted data augmentation and model tuning.

Confidence Thresholding

To ensure reliability in the field, predictions with a confidence score below **60%** are categorized as "Unknown". This minimizes false positives and guides farmers to seek manual verification.

Installation & Usage

Setup

Clone the repository and install the required dependencies in a virtual environment.

```
git clone repo_url  
pip install -r requirements.txt
```

Running the App

Launch the Gradio web interface for real-time inference testing.

```
python app.py
```

*Opens local web server for image upload

”The next phase involves deploying this lightweight model to edge devices for offline, real-time video analysis in the field.”

Future Roadmap

Q & A

Thank you for reviewing this assignment.