**Smart Agriculture: Crop Disease Detection**

**1. Objective**

The primary goal of this project is to build an AI-powered system that detects plant diseases from leaf images using deep learning. This tool can assist farmers and agronomists in:

* Early identification of diseases
* Reducing crop loss
* Increasing yield through timely intervention

**2. Dataset Overview**

**Source:** CCMT Plant Disease Dataset

* **Total Images**: 25220
* **Categories**: 22 Disease classes of Ghanaian crops (cashew, cassava, maize, tomato)
* **Classes include**:
  + **Healthy leaves**
  + **Diseased leaves** of 21 clasess

**Image Format**: RGB .jpg  
**Image Size**: Varied; resized to 224x224 for uniform input to CNN

**3. Dataset Generation and Data Preprocessing**

Initially I have split the entire dataset to two folders, one containing only healthy plant images and other containing all the diseased images so that it will easy to read, add respective labels and to create the dataset.

Removed all the bad images and filtered fully valid images to get the final dataset. The dataset sized reduced from 25220 to 25126.

* **Resizing** all images to 224x224 using TensorFlow/Keras preprocessing utilities
* **Label Encoding**: Converted string labels to one-hot vectors

**4. Model Architecture**

This project uses MobileNetV2, a lightweight convolutional neural network architecture optimized for mobile and edge devices. It's pre-trained on ImageNet and fine-tuned on the plant disease dataset.

Why MobileNetV2?

* Lightweight & efficient: Suitable for real-time, mobile deployment
* Transfer learning: Leverages pre-trained visual features
* Good generalization with relatively few training samples

**5. Training Configuration**

* Epochs: 10
* Optimizer: Adam (learning rate: 1e-4)
* Loss Function: Categorical Crossentropy
* Input Image Size: 224x224x3
* Batch Size: 32
* Callbacks:
  + EarlyStopping(monitor='val\_loss', patience=3)
  + ModelCheckpoint(filepath='plant\_disease\_model.h5', save\_best\_only=True)

**6. Evaluation Metrics**

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 76.9% |
| Validation Accuracy | 77% |
| Loss | 64.9% |

**7. 🌐 Deployment**

**Frontend: HTML and CSS**

**Features**:

* + Upload an image
  + Display uploaded image
  + Display top prediction with confidence score

**Backend: FASTAPI**

* .h5 model loaded with TensorFlow
* Image preprocessing using PIL and NumPy
* Prediction

**8. Challenges and Solutions**

| **Challenge** | **Description** | **Solution** |
| --- | --- | --- |
| **Images Filtering** | Some images were bad and had difficulty while training. | Used UnidentifiedImageError to identify good images. |
| **Accuracy** | Initial accuracy was low. | Increase the epoch value to get a good accuracy. |

**9. Deliverables**

| **File** | **Description** |
| --- | --- |
| plant\_disease\_model.h5 | Final trained deep learning model |
| app.py | Using FastAPI (Backend) |
| Index.html & Style.css | Front end |
| requirements.txt | List of Python dependencies |
| README.md | Instructions and setup |
| documentation.pdf | This full project report |

**10. Conclusion and Future Work**

* The system accurately identifies 21 plant diseases with over 77% test accuracy.
* Easily extensible to support more classes or new crops.
* Real-time predictions via a simple UI can assist farmers with minimal effort.

**Future Improvements:**

* Train with real-world, noisy, low-resolution images
* Integrate into a mobile app using TensorFlow Lite
* Enable geo-tagging and disease tracking over time
* Add explainability with Grad-CAM visualizations