AI for AgriTech Hackathon - Stage 1 Submission

# 1. Project Title

AI-Based Plant Disease Detection using CNN

NAME: THITHIKSHA.P.K

COLLEGE: SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

DEPARTMENT: B.TECH INFORMATION TECHNOLOGY

2. Introduction

This project explores how Artificial Intelligence (AI) and Computer Vision (CV) can be used to detect plant diseases from images of leaves. The goal is to help farmers identify diseases at an early stage and take corrective actions.

# 3. Dataset Description

The dataset used in this project includes manually uploaded images of tomato leaves. Three classes were created:  
- Healthy  
- Bacterial Spot  
- Late Blight  
Each class contains around 10 images.

# 4. Model Architecture

The model is based on a Convolutional Neural Network (CNN) architecture automatically generated by Google Teachable Machine. It includes the following layers:  
- Input Layer (224x224 RGB image)  
- Convolution + ReLU  
- Pooling  
- Fully Connected Layers  
- Softmax Output Layer (3 classes)

# 5. Why CNN Was Chosen

CNNs are the most effective architecture for image classification problems. They automatically learn features such as edges, textures, and patterns in leaves, which are essential for identifying plant diseases.

# 6. Training Process

The model was trained using Google Teachable Machine. Around 30 images (10 per class) were used for training. The training process completed successfully with high accuracy.

# 7. Evaluation Metrics

Teachable Machine does not provide all metrics, but the following were observed:  
- Accuracy: ~95%  
- Precision and Recall: High (manually observed)  
- Other metrics such as IoU, mAP, SSIM, PSNR, and MSE are not available in the interface and can be evaluated externally.

# 8. Optimization Techniques

To improve performance:  
- Images were resized and centered  
- Class balance was maintained  
- The model was exported in Keras format for future fine-tuning

# 9. Exported Model

The trained model was exported as a Keras model in `.h5` format and saved as a `.zip` file. This file is attached for evaluation.

# 10. Screenshots

of the training process, prediction results, and export screen are to be added here. Placeholders can be used if screenshots are not available.

