

# PLANT DISEASE DETECTION USING DEEP LEARNING

---

## Introduction

### Objective:

The goal of this project is to develop a deep learning-based model for plant disease detection using images of plant leaves. The trained model will be deployed in a Streamlit-based web application, where users can upload leaf images to receive predictions about potential diseases.

This study leverages CNNs (Convolutional Neural Networks) and transfer learning with pre-trained models such as EfficientNetB0, ResNet50, and DenseNet121, optimizing performance using data augmentation, mixed precision training, and learning rate adjustments.

---

## Dataset Description

### Dataset Source:

The dataset used is the New Plant Diseases Dataset (Augmented) from Kaggle, containing labeled images of plant leaves affected by different diseases.

### Preprocessing Steps:

- **Resizing:** All images resized to 128×128 pixels for compatibility with the models.
  - **Augmentation:** Random transformations applied to enhance generalization, including:
    - Rotation
    - Width & Height Shift
    - Zoom & Shear
    - Horizontal Flip
    - Brightness
- 

## Model Architecture and Training

### Pre-trained Models Used (Transfer Learning)

Three deep learning models were fine-tuned for plant disease classification:

1. **EfficientNetB0**
2. **ResNet50**
3. **DenseNet121**

Each model was initialized with ImageNet weights and modified to:

- Include Global Average Pooling, Dense, and Dropout layers.
- Replace the classifier with a fully connected layer matching the number of plant disease classes.

---

## Training Configuration

### Hyperparameters:

Parameter	Value
Batch Size	32
Epochs	10
Optimizer	Adam (learning rate = 0.001)
Loss Function	categorical_crossentropy

### Training Strategy:

- **Early Stopping:** Prevents overfitting by monitoring validation loss.
- 

## Performance Evaluation

Each model was evaluated using the validation dataset based on accuracy, precision, recall, and F1-score.

### Results Summary

Model	Accuracy	Loss
EfficientNetB0	0.9301	0.1999
ResNet50	0.9292	0.2005
DenseNet121	0.9011	0.2881

**Best Model: ResNet50**(Considering the model architecture and above mention performances)

---

## Deployment & Application Features

A Streamlit-based web app was developed to allow users to upload images and receive real-time predictions.

### Features:

- **Image Upload:** Accepts JPEG, PNG, and BMP files.
  - **Model Selection:** Users can choose between EfficientNetB0, ResNet50, and DenseNet121.
  - **Real-Time Prediction:** Model inference runs instantly on uploaded images.
  - **Disease Interpretation:** Displays predicted class and a confidence score.
- 

## Conclusion & Future Work

### Key Findings:

- EfficientNetB0 achieved the highest accuracy (93.01%) .
- ResNet50 (92.92%) outperformed other models with simpler architectures benefit from transfer learning in plant disease detection and demonstrated superior generalization.

- Transfer learning improved performance significantly, reducing the training time while maintaining high accuracy.

#### **Future Enhancements:**

- Train models on higher resolution images (e.g., 224×224) for better feature extraction.
- Experiment with self-supervised learning techniques for label-efficient training.
- Optimize the web application UI/UX for better user experience.
- Expand the dataset to include more plant species and diseases for broader applicability.

---

#### **Deliverables**

- **Trained Models:** EfficientNetB0\_.h5, ResNet50\_.h5, DenseNet121\_.h5.
  - **Streamlit Application:** Plant Disease Detection Web App.
  - **Codebase:** Python scripts for model training, preprocessing, and deployment.
  - **Project Report:** This document summarizing results, methodology, and future scope.
-