PROJECT ABSTRACT

Plant Disease Detection using CNN (PlantVillage)

A ResNet-50 transfer-learning classifier for 38 crop-disease classes on the PlantVillage dataset, with a leaf-diagnosis web demo. Delivered built-to-order with training notebook, model, report, PPT and viva kit.

---

01. Title

Plant Disease Detection using CNN (PlantVillage)

02. Introduction / Background

Crop diseases destroy a significant share of yields every season, and early visual diagnosis remains the main line of defense for most farmers. Plant pathologists cannot be everywhere, so an automated leaf-image classifier that runs on a phone photo is genuinely useful. Convolutional neural networks excel at fine-grained visual discrimination, and transfer learning makes it practical to train a 38-class classifier from a public dataset. This project fine-tunes a ResNet-50 backbone on the full PlantVillage corpus, augments training with field-condition transforms, evaluates on a held-out test split with per-class metrics, and delivers a working web demo that classifies uploaded leaf photos with confidence scores and treatment guidance.

03. Problem Statement

Most student builds of plant-disease classifiers train on tiny, clean datasets and collapse on real field photos, and they rarely document why each design choice was made — leaving the student unable to defend the work in a viva. The project addresses this by training on the full 54,305-image PlantVillage corpus with field-condition augmentation, a stratified held-out evaluation, and a documented architecture rationale, plus a demo whose outputs are agronomically useful rather than decorative.

04. Objectives

- Fine-tune ResNet-50 to classify 38 PlantVillage crop-disease classes from leaf photos.
- Apply field-condition augmentation (rotation, brightness, blur) for robustness to phone-camera photos.
- Evaluate on a held-out test split with per-class precision, recall, F1 and a confusion matrix.
- Build a web demo that classifies uploads and returns treatment guidance per predicted class.
- Document every design choice for viva defense.

05. Proposed Methodology

The project is executed in four stages. First, the PlantVillage dataset is loaded with stratified train/validation/test splits and a 256×256 preprocessing pipeline with ImageNet normalization.

Second, a ResNet-50 backbone pretrained on ImageNet is adapted with a 38-class head; the head trains with the backbone frozen, then top blocks are unfrozen for fine-tuning. Third, training runs up to 40 epochs with early stopping while augmentation simulates field conditions. Fourth, the held-out test set is evaluated once and all metrics, curves and the confusion matrix are exported for the report and demo.

06. System Architecture / Working

- Data layer: 54,305 PlantVillage images, stratified splits, augmentation pipeline.
- Model: ResNet-50 backbone + custom 38-class head with dropout.
- Training: two-phase fine-tuning, Adam optimizer, early stopping on validation accuracy.
- Evaluation: held-out test accuracy, per-class F1, confusion matrix.
- Demo: upload → identical preprocessing → saved model → prediction + confidence bars + treatment guidance.

07. Software Requirements

- Python 3 with TensorFlow/Keras, NumPy, Matplotlib, scikit-learn.
- Jupyter Notebook for training and evaluation.
- A free cloud GPU (e.g. Colab) for the 2–4 hour training run; CPU suffices for inference.
- Any modern web browser for the diagnosis demo — no server required.

08. Expected Outcomes

A trained 38-class classifier with a design-target test accuracy of ~94% (documented honestly after the training run), a working leaf-diagnosis web demo, per-class metrics and confusion matrix, training-curve plots, and a complete report/PPT/viva kit. The student can explain transfer learning, augmentation and imbalanced evaluation from first principles.

09. Applications

- Early crop-disease screening from phone photos for farmers and extension workers.
- Agricultural advisory apps and agri-tech prototypes.
- Teaching transfer learning and fine-grained image classification.

10. Future Scope

- Leaf segmentation stage for cluttered field backgrounds.
- Severity estimation (percentage leaf area affected) as a regression head.
- On-device deployment via TensorFlow Lite for offline farm use.