Agricultural Pest Detection

Course Title: Computer Vision

Submitted To:

Sir Mohammad Rizwan

Submitted By:

- **Student 1: Ahmad Yasin** **L1F22BSAI0052**
- **Student 2: Eman Sarfraz** **L1F22BSAI0034**
- **Student 3: Mariam Tariq** **L1F22BSAI0018**
- **Student 4: Abdul Rehman** **L1F22BSAI0031**

Section: BSAI – AF1

1. Introduction

The agricultural sector faces significant threats from pest infestations, which negatively affect crop productivity and food security. Timely and precise identification of pests is crucial for effective intervention and pest management. Nevertheless, conventional detection methods tend to be manual, labour-intensive, and prone to errors, particularly in extensive agricultural settings.

This project introduces a deep learning-based approach for automatic pest classification utilizing image data. We employ transfer learning with EfficientNet variants (B0–B3) to create robust classification models using the publicly accessible Agricultural Pests Image Dataset from Kaggle. By fine-tuning pre-trained models, we enhance accuracy while minimizing the requirement for extensive training datasets.

To facilitate practical application, we also create a comprehensive web application. The frontend is developed using **React.js/Next.js** and deployed on **Vercel**, while the backend is constructed in **Flask** and hosted on **PythonAnywhere**. This architecture guarantees a swift, responsive, and easily accessible interface for farmers and agricultural experts.

2. Objectives

- Develop a high accuracy pest detection model using image classification.
- Implement transfer learning using EfficientNet B0 to B3 for better performance and efficiency.
- Fine-tune models.
- Achieve a minimum test accuracy of 94%.
- Compare performance across EfficientNet variants to assess accuracy vs. computational cost.
- Build and deploy a full-stack web application for real-time pest detection.
- Ensure the solution is scalable and user-friendly for practical deployment in agriculture.

3. Methodology

Dataset

- **Name:** Agricultural Pests Image Dataset
- **Source:** Kaggle
- **Description:** Contains labelled images from 12 pest categories captured in real-world agricultural settings.
- **Preprocessing:** Resizing, normalization, and augmentation

Modelling Approach

- **Pre-trained Models:**
  - EfficientNet-B0
  - EfficientNet-B1
  - EfficientNet-B2
  - EfficientNet-B3
- **Techniques Used:**
  - Replace classification layers with custom heads suitable for 12-class output
  - Fine-tune deeper layers based on validation performance
  - Regularization through dropout, learning rate scheduling, and data augmentation
  - Hyperparameter tuning integrated via transfer learning strategy

Development Stack

Component	**Technology**
Frontend	React.js / Next.js
Frontend Hosting	Vercel
Backend	Flask
Backend Hosting	PythonAnywhere
ML Libraries	PyTorch, torchvision, sklearn, matplotlib, seaborn
Languages	Python, JavaScript

Pipeline Process

1. Load and preprocess dataset.
2. Select EfficientNet variant and customize classifier head.
3. Fine-tune using transfer learning on the training set.
4. Evaluate using validation and test splits (accuracy, loss, confusion matrix).
5. Save the best model and integrate it into Flask backend.
6. Build a responsive UI with React.js/Next.js for file upload and results display.
7. Deploy frontend on Vercel and backend on PythonAnywhere.
8. Perform end-to-end testing of the live application.

4. Expected Outcomes

- High performing pest detection models achieving 94% accuracy.
- Comparative evaluation of EfficientNet-B0 to B3 in terms of performance and efficiency.
- Deployment-ready full-stack solution accessible via browser.
- Improved generalization and reduced overfitting through model regularization.
- A portable and reusable system for agricultural image classification tasks.

5. Timeline

Week	Milestone
Week 1	Dataset analysis and model setup
Week 2	Model fine-tuning and training (EfficientNet B0–B3)
Week 3	Performance evaluation and backend API development
Week 4	Frontend integration, deployment, testing, and report

6. References

1. Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* ICML.
2. Kaggle Dataset: Agricultural Pests Image Dataset – `https://www.kaggle.com/datasets/vencerlanz09/agricultural-pests-image-dataset`
3. PyTorch Documentation – `https://pytorch.org/docs/`
4. Flask Documentation – `https://flask.palletsprojects.com/`
5. React.js – `https://reactjs.org/`
6. Vercel Platform – `https://vercel.com/`
7. PythonAnywhere – `https://www.pythonanywhere.com/`