Project Report:

Title: Multi-Crop Plant Recognition for Smart Agriculture Using Deep Learning

---

Abstract: Precision agriculture relies on timely and accurate identification of crops for optimizing yield, monitoring biodiversity, and enabling automated farm management. Manual crop identification is slow, error-prone, and requires expert knowledge, which is often unavailable in rural areas. This project proposes a scalable deep learning-based solution capable of recognizing 139 crop types from field images. The system uses MobileNetV2 with transfer learning and a Streamlit interface for mobile and web accessibility. Grad-CAM is implemented for model explainability, and an optional LLM integration provides agronomic advice for farmers.

---

1. Introduction: Agriculture faces challenges in crop monitoring due to varied plant species, environmental conditions, and lack of experts in remote regions. Traditional methods are labor-intensive and not scalable. AI-based image classification presents a solution for real-time crop recognition, enhancing efficiency and reducing manual errors. The objective of this project is to develop a deep learning model capable of identifying 139 crop species from images and deploy it in a user-friendly web/mobile interface.

Objectives: - Train a deep learning model to classify 139 crop categories. - Achieve a classification accuracy $\geq$90%. - Ensure robustness to variations in lighting, background, and image quality. - Deploy a mobile/web application for image classification. - Optional: Integrate LLM to provide agronomic advice based on predicted crops.

---

2. Literature Review: - **Convolutional Neural Networks (CNNs):** Core architecture for image classification. - **ResNet/VGG:** Deep CNN models for accurate feature extraction. - **MobileNet/EfficientNet:** Lightweight models optimized for mobile deployment. - **Transfer Learning:** Reduces training time and improves performance using pre-trained weights. - **Agricultural AI Applications:** Prior works mostly cover limited crop species or lack deployment considerations.

Datasets: - Kaggle 139 Crops Image Dataset (RGB 224x224 images, >100,000 images).

Tools & Libraries: - TensorFlow, Keras, PyTorch - OpenCV, PIL - FastAI - Streamlit for frontend - SHAP and Grad-CAM for explainability - OpenAI API for LLM integration

---

3. Methodology: Data Preprocessing: - Resize images to 224x224 pixels. - Normalize pixel values to [0,1]. - Apply augmentation: horizontal flip, rotation, zoom.

Exploratory Data Analysis (EDA): - Class distribution plotted (training set: 27,639 images, validation set: 6,422 images, test set: 6,627 images). - Visual inspection of sample images showed variability across species.

Model Architecture: - Base model: MobileNetV2 pre-trained on ImageNet. - Added GlobalAveragePooling, Dense layers, Dropout, and final softmax layer. - Training optimizer: Adam with initial learning rate 1e-3. - Fine-tuning performed by unfreezing base layers with learning rate 1e-5.

Training: - Initial training for 30 epochs achieved top-1 validation accuracy starting at 31% (Epoch 1) and gradually improving to ~53%. - Fine-tuning for 10 epochs further improved performance to ~53–54% validation accuracy. - Observed loss decrease over epochs indicated model learning progression.

Evaluation Metrics: - Overall validation accuracy: 54% - Macro average F1-score: 0.52 - Weighted average F1-score: 0.54 - Class-wise performance: Precision ranged from 0.39 (Barley) to 0.82 (Carrots and turnips), recall ranged from 0.33 (Camucamu) to 0.90 (Artichoke), showing class imbalance effects. - Confusion matrix plotted to visualize misclassifications.

Explainability: - Grad-CAM applied to test images highlighting regions important for predictions. - Heatmaps confirmed the model focused on leaf structures and plant-specific features.

Deployment: - Streamlit-based web app allows image upload and real-time prediction. - Predicted class displayed with confidence score. - Grad-CAM heatmap visualization button included. - Optional button to query LLM for agronomic advice using predicted crop.

---

4. Results: - Training/Validation accuracy and loss trends indicate effective transfer learning and fine-tuning. - Class distribution and sample image visualization ensured dataset comprehensiveness. - Grad-CAM results provide interpretability for model decisions. - LLM integration successfully provides textual agronomic advice based on predicted crops. - Real-time predictions in the app validated model usability in field-like scenarios.

Limitations: - Accuracy below desired 90% due to 139-class complexity and dataset imbalance. - Offline functionality not implemented. - Performance sensitive to image quality and lighting conditions.

---

5. Discussion: - MobileNetV2 with transfer learning enabled efficient training and deployment on web/mobile. - Class-wise variability in F1-score indicates need for additional balanced data. - Grad-CAM provides visual explanations enhancing trust. - LLM integration offers practical value for farmers by generating care tips.

Future Work: - Expand dataset with more diverse images. - Optimize model for offline/mobile usage (TensorFlow Lite). - Incorporate plant disease detection. - Explore larger CNNs or ensemble methods for higher accuracy. - Multi-lingual LLM advice for broader accessibility.

---

6. Conclusion: This project demonstrates a deep learning-based multi-crop recognition system capable of predicting 139 crop types. The model, using MobileNetV2 and transfer learning, achieved ~54% validation accuracy. The system is deployed as a Streamlit web/mobile app with Grad-CAM explainability and optional LLM agronomic advice. While accuracy can be improved, the project lays a foundation for scalable smart agriculture solutions.

---

7. References: 1. Howard, A. G., et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv:1704.04861, 2017. 2. Tan, M., et al. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." ICML 2019. 3. He, K., et al. "Deep Residual Learning for Image Recognition." CVPR 2016. 4. Kaggle 139 Crops Image Dataset, https://www.kaggle.com/datasets. 5. Chollet, F. "Deep Learning with Python." Manning Publications, 2017. 6. TensorFlow/Keras

Documentation, https://www.tensorflow.org. 7. OpenAI API Documentation, https://platform.openai.com/docs/

---

8. Appendices: - Training and validation accuracy/loss plots. - Sample predictions with Grad-CAM overlays. - Confusion matrix visualization. - Screenshots of Streamlit app interface. - Selected code snippets for reproducibility.