AI-Based Smart Agriculture

System for Crop Health Monitoring

Tomiwa Adeyemi
BEng Robotics and Artificial Intelligence
Supervisor: Zakaria Chekata
University of Hertfordshire 27 March 2026

Project Defence

Introduction & Problem Statement

Robustness Gap

Models trained on controlled PlantVillage images drop from 99% → 31% accuracy in real field conditions (Mohanty et al., 2016)

Model Size

VGG-16 (138M params) and ResNet-50 (25M params) are too large for battery-powered smartphones used by smallholder farmers

Black-Box Problem

High validation accuracy provides no spatial evidence that lesions not background artefacts are the driving predictions

Plant diseases cause 20–40% of global crop yield losses annually, costing over $220 billion (FAO, 2022)

Aims & Objectives

AIM — Design, evaluate and compress a deep learning model for 38-class plant disease classification, optimised for edge device deployment whilst preventing catastrophic forgetting.

01	**Robustness Pipeline**	Grayscale conversion + Gaussian noise (σ=0.05) to reduce colour dependency
02	**Architectural Evaluation**	Custom Baseline CNN vs MobileNetV2 under identical training conditions
03	**Spatial Interpretability**	Grad-CAM heatmaps to verify lesion-focused feature learning
04	**Edge Optimisation**	Layer-wise L1 unstructured pruning at approximately 15% sparsity
05	**Performance Recovery**	Fine-tuning at lr=0.0001 post-pruning; evaluate via confusion matrices & F1

Methodology

Dataset	Preprocess	Train	Explain	Prune	Recover
PlantVillage 54,305 images 38 classes	Grayscale + AWGN noise 70/15/15 split	Baseline CNN vs MobileNetV2 20 epochs	Grad-CAM Saliency maps both models	L1 unstructured 14.67% sparsity layer-wise	Fine-tune lr=0.0001 5 epochs

Key Design Decisions

Why MobileNetV2?

Approximately 3.5M parameters; 93% fewer than Baseline CNN. ImageNet pre-trained backbone with frozen weights; only the 38-class head trained from scratch.

Why layer-wise pruning?

Protects first conv (low-level features) and final classifier (38-way boundary). Avoids catastrophic forgetting from naïve global pruning.

Why grayscale + noise?

Reduces reliance on colour cues that vary with lighting and camera type. Simulates real-world sensor degradation without full field dataset.

Reproducibility controls

Global seed=42 across Python, NumPy, PyTorch and CUDA. cuDNN deterministic mode. Seeded DataLoader generator.

Results — Training & Explainability

88.31%
MobileNetV2
Test Accuracy

82.97%
Baseline CNN
Test Accuracy

+5.34pp
Transfer learning
advantage

Grad-CAM Spatial Validation

MobileNetV2

Smooth, anatomically coherent heatmaps concentrated on leaf blade and central venation. No activation in background regions.

Baseline CNN

Sparse, punctate hotspots on texture artefacts. Fragmented saliency model attends to noise rather than disease lesion morphology.

MobileNetV2's interpretable, lesion-focused activations confirmed it as the architecturally sound choice for deployment — not just the most accurate

Results — Compression & Recovery

Pre-pruning
88.31%
MobileNetV2 baseline

→

Post-pruning
60.58%
−27.73pp | intentional stress test

→

After fine-tuning
97.56%
+9.25pp above original

328,329
Weights pruned

2,238,400
Eligible weights

14.67%
Sparsity achieved

5 epochs
Recovery duration

Class Fairness Improvement

Class	Before	After
Macro F1	0.84	0.97
Tomato Early Blight F1	0.50	0.87
Tomato Leaf Mold F1	0.65	0.94
Tomato Target Spot F1	0.69	0.94

Critical Reflection & Limitations

What went well	Key limitations
Reproducible seed-controlled pipeline produced consistent results across all training runs	Single-split, single-run design: 97.56% is a strong observed result, not a statistically validated population estimate
MobileNetV2 post-pruning gain of +9.25pp was unexpected (pruning acted as implicit regularisation)	Integrity check applied to binary reorganisation, not directly to the 38-class training split
Grad-CAM maps consistently localised to leaf tissue, providing spatial validation beyond accuracy metrics	Unstructured pruning reduces model size but does not deliver real-world latency gains on standard hardware
All four research questions answered with empirical evidence	Grayscale trade-off: colour cues that genuinely distinguish tomato disease sub-classes were removed

Conclusions & Future Work

1
Deployable pipeline

Reproducible PyTorch pipeline with seed control, stratified splitting, and modular training infrastructure

2
Architecture validated

MobileNetV2 (88.31%) outperforms Baseline CNN (82.97%)

3
Compression proven

14.67% sparsity with recovery to 97.56% with a macro F1 0.97 across all 38 classes post fine-tuning

4
Explainability verified

Grad-CAM confirms predictions grounded in leaf tissue, not background artefacts (deployable with stakeholder trust)

Next steps for publication-ready work

- 5-fold cross-validation across multiple seeds
- Hardware profiling on Raspberry Pi / Jetson Nano
- Out-of-distribution testing on PlantDoc field images

Thank you — Questions welcome