



Multi-Crop Plant Recognition Pipeline

End-to-end production-ready system: 139 plant species classifier
with Streamlit web interface and AI-powered agronomy guidance

Radhey Pareek



Problem Statement & Solution

Challenge

Farmers need rapid, accurate plant species identification with actionable agronomy guidance for diseases, remedies, climate requirements, and nutrients.

Solution

139-class deep learning classifier with robust inference pipeline featuring Test Time Augmentation (TTA), multi-crop averaging, and bilingual AI guidance via Gemini integration.

- ✓ **Key Innovation:** Production-optimized pipeline balancing accuracy with speed for modest hardware deployments



Technical Architecture Overview



Data Pipeline

Raw images \Rightarrow RGB_224x224 preprocessing with ImageNet normalization and robust imread handling



Model Training

EfficientNet-B0 backbone with AdamW optimizer, cosine scheduling, and mixed precision training



Inference Engine

TTA + 5-crop averaging for robust predictions with configurable confidence thresholds



Web Application

Streamlit UI with camera/upload support and integrated Gemini AI agronomy guidance

Technology Stack & Dependencies

Deep Learning Core

- PyTorch + torchvision
- timm model zoo
- Mixed precision (AMP)

Data Processing

- Albumentations (fast augmentation)
- OpenCV + Pillow
- Hydra configuration

Production Interface

- Streamlit web UI
- Google Gemini API
- Minimal runtime deps

Rationale: [timm](#) provides battle-tested backbones, [Albumentations](#) offers fast transforms, and [Streamlit](#) accelerates demo-ready deployment.

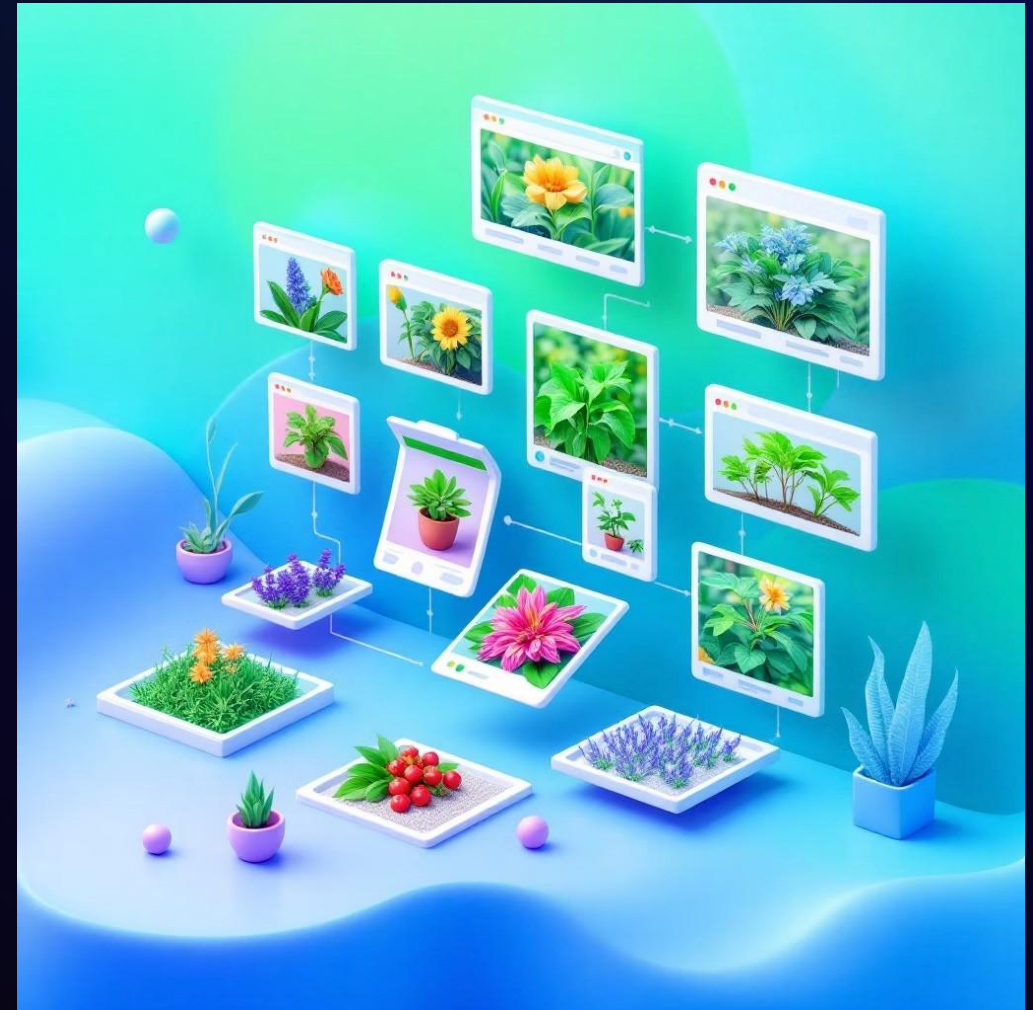
Data Pipeline & Processing

Dataset Structure

```
data/raw/<class>/*.jpg
⇒ (prepare_dataset.py)
data/RGB_224x224/
=%% train/ (70%)
=%% val/ (15%)
5%% test/ (15%)
```

Key Features

- Robust `imread_rgb()` handles grayscale, RGB, RGBA
- Stable `class_to_idx` mapping preservation
- ImageNet mean/std normalization
- Training at 160-192px for speed optimization



Speed Optimization: Smaller image sizes and reduced validation frequency (every 4 epochs) for faster training cycles



Model Configuration & Training Strategy

1

Architecture

Backbone: EfficientNet-B0 (default) / MobileNetV2 (speed profile)

Classes: 139 plant species

Input: 192×192 RGB images

2

Optimization

Optimizer: AdamW (lr=1e-3, wd=0.01)

Scheduler: Cosine annealing

Precision: Mixed precision + channels-last

3

Training Setup

Batch size: 48 | **Epochs:** 30

Augmentation: Flips, rotations, color jitter, dropout

Validation: Every 4 epochs for efficiency

Robust Inference Engine

InferenceModel Features

- Auto-loads `weights/best.pt` with embedded config
- Preserves `class_to_idx` mapping
- Configurable TTA modes
- Multi-crop averaging support

TTA Strategies

- **Off:** Original image only
- **Fast:** + horizontal flip
- **Full:** + vertical flip

01

5-Crop Extraction

Center + 4 corners for robust framing coverage

02

TTA Application

Apply augmentations to each crop independently

03

Prediction Averaging

Ensemble predictions across all crops and augmentations

04

Confidence Thresholding

Auto-retry with stronger TTA if confidence < 0.35

Streamlit Web Application



Image Input

Supports file upload and real-time camera capture with preprocessing pipeline integration



Smart Inference

Cached model loading, multi-crop predictions, and adaptive TTA based on confidence scores



AI Guidance

Bilingual agronomy advice (English/Hindi) via Gemini API with structured output formatting

Green-themed responsive design with animated background and graceful fallbacks for offline scenarios



Production Results & Performance

Model Checkpoints

Best model: `weights/best.pt`

Validation: Every 4th epoch

Metrics: Embedded in
checkpoint

```
import torch
ckpt =
torch.load('weights/best.
pt')
print(ckpt['best_metric'])
```

139

Plant Classes

Comprehensive multi-
crop species coverage

192

Input Size

Optimized resolution for
speed-accuracy balance

5

Crop Ensemble

Center + corners for
robust predictions

30

Training Epochs

Efficient convergence
with cosine scheduling



Key Achievement: Production-ready pipeline with minimal dependencies and embedded configuration for consistent deployment



Deployment & Future Roadmap

1

Current State

- 154-class classifier
- Streamlit web interface
- Gemini AI integration
- Production optimizations

2

Near Term

- Evaluation metrics script
- Early stopping & EMA
- ONNX export for edge
- Grad-CAM visualizations

3

Long Term

- 🎯 Multi-language expansion
- 🎯 Offline AI fallbacks
- 🎯 TensorRT optimization
- 🎯 Class imbalance handling

Ready to Deploy: `streamlit run app.py` with Gemini API key for immediate agricultural AI assistance