Fruit and Vegetable Disease Detection System

Leveraging machine learning and computer vision to automate early disease detection and promote healthier, more productive crops.

This innovative system enables farmers to identify diseases at their earliest stages, reducing crop losses and minimizing the use of harmful pesticides. By providing real-time analysis and actionable insights, it supports sustainable agriculture practices and improves overall yield quality.

Project Overview

Objective

Detect diseases in fruits and vegetables accurately.

Approach

Use computer vision techniques powered by PyTorch.

Data Source

Public datasets like Kaggle's "Healthy vs. Rotten Fruits."

Problem Domain

Agricultural automation and food quality control focus.

Data Acquisition and Preprocessing

Dataset Sources

- Kaggle and PlantVillage
- Field-collected fruit/vegetable images

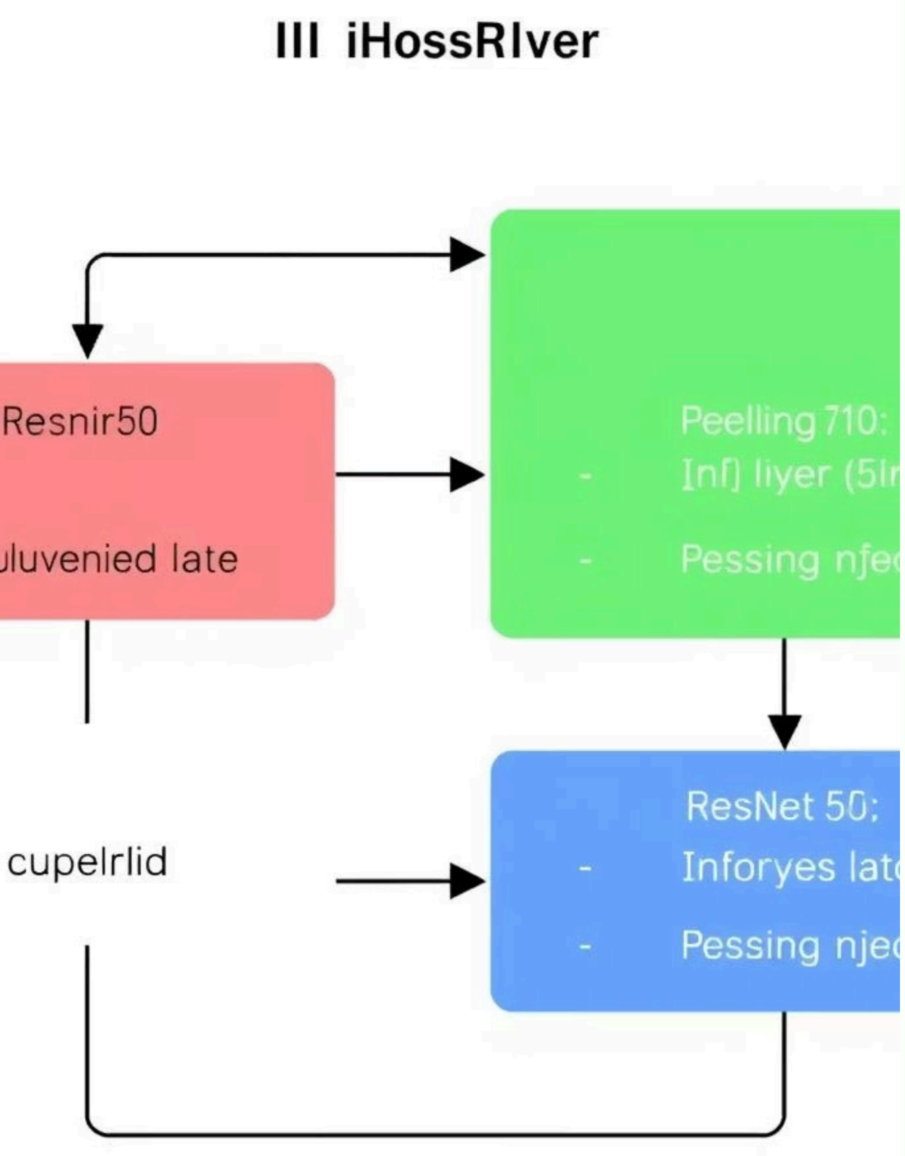
Data Types

RGB images resized to 256x256 pixels

Preprocessing Steps

- Normalization with ImageNet stats
- Data augmentation: rotation, flips, zooms
- Split: 80% train, 10% val, 10% test

Labels verified; labeling cost $0.04 per image



Model Architecture

Base Model

ResNet50 pre-trained on ImageNet for feature learning

Custom Layers

Adaptive pooling, fully connected layers, ReLU, dropout 0.5

Optimization

Cross-entropy loss and Adam optimizer, LR 0.001, weight decay 0.0001

Training and Validation

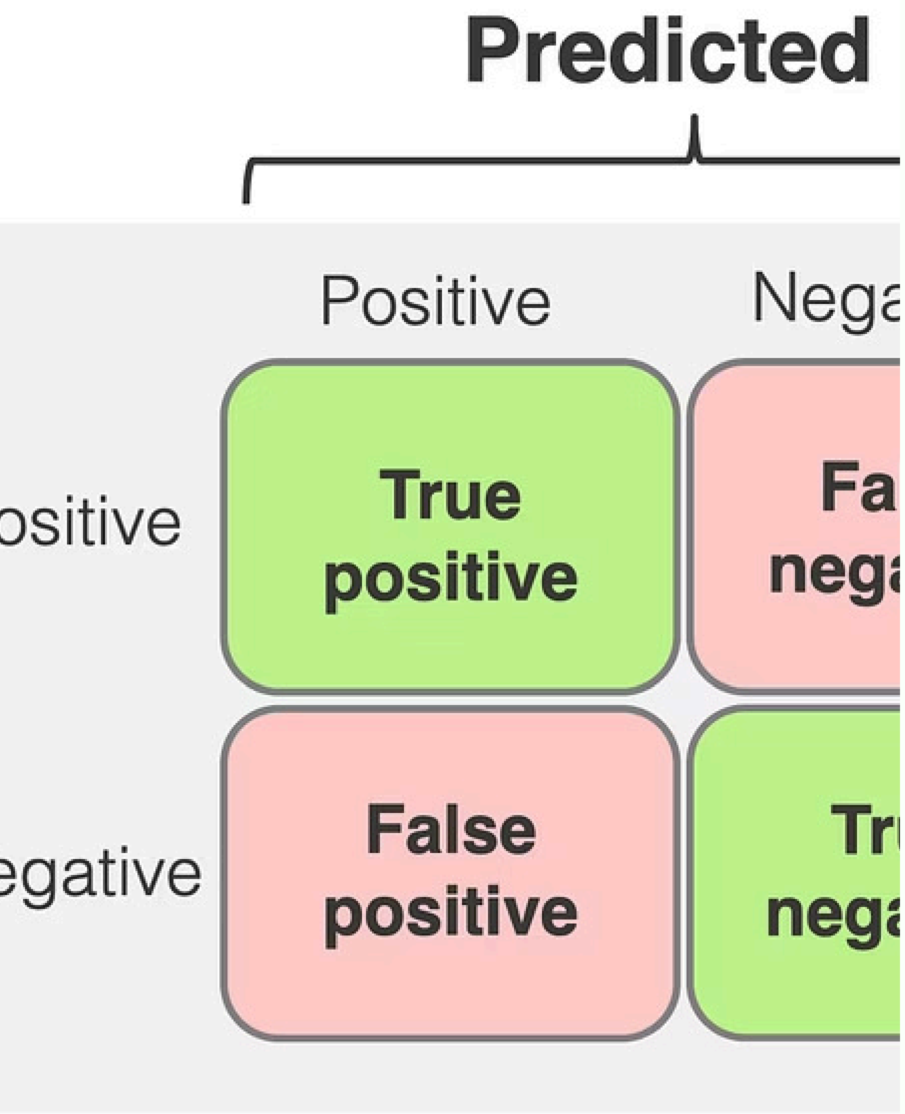
Environment & Hardware

- PyTorch 1.10, CUDA 11.3, Python 3.8
- NVIDIA Tesla V100 GPU with 16GB memory

Training Settings

- Batch size 64, 50 epochs
- Learning rate decay on plateau
- Metrics: Accuracy, Precision, Recall, F1-Score

Achieved 92% validation accuracy, overfitting controlled via L2 regularization



Testing and Evaluation

Test Dataset

Used unseen data to validate generalization

Evaluation Metrics

- Overall accuracy: 91%
- Precision: 92% weighted average
- Recall: 90% weighted average
- F1-score: 91% weighted average

Analysis

Confusion matrix and class-specific performance insights

Deployment Strategy

Platform

AWS SageMaker for scalable deployment

Model Optimization

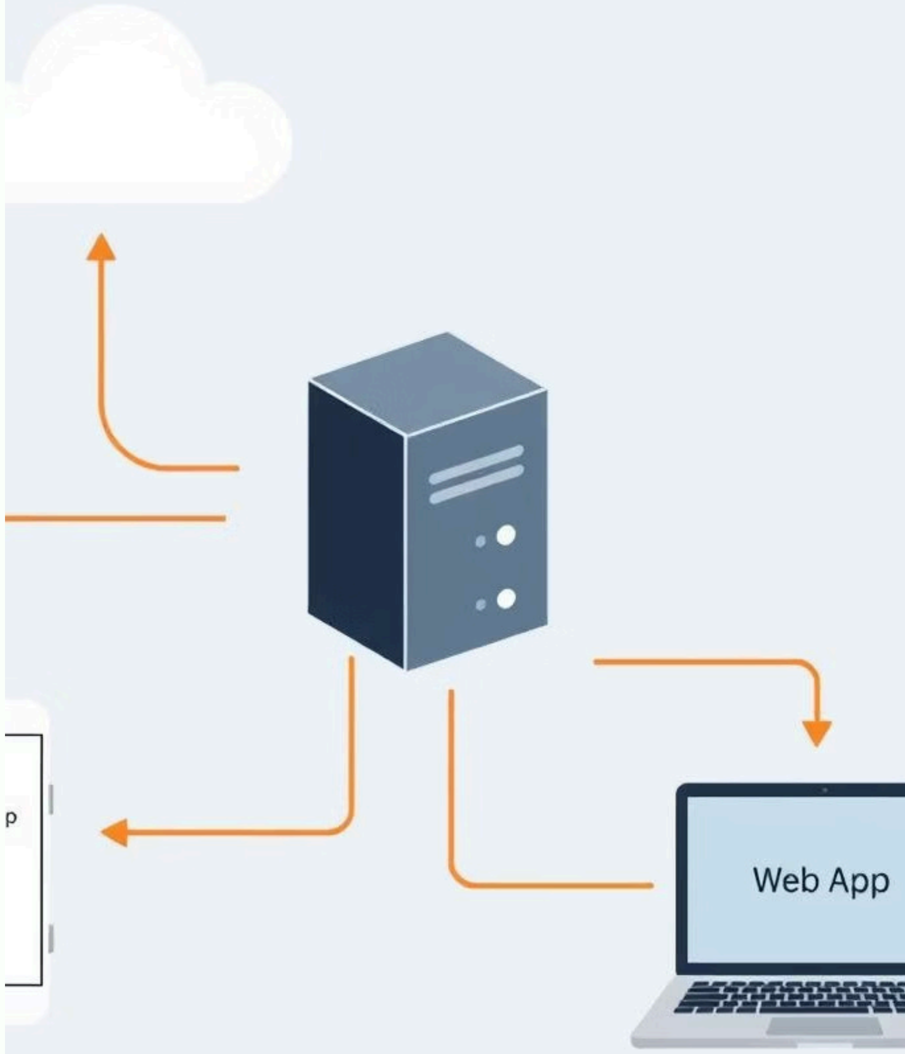
Quantization and pruning cut model size by 50%

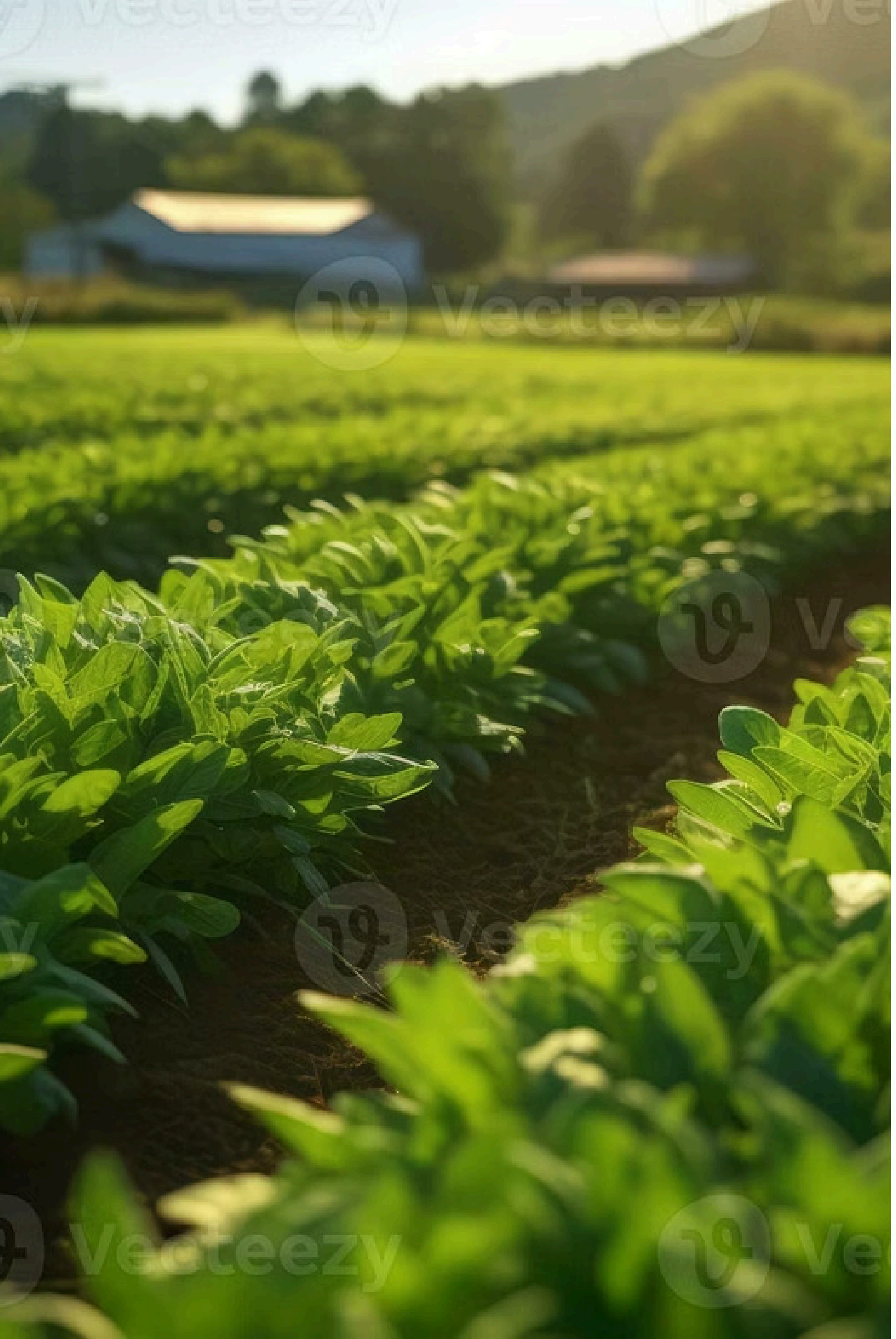
API & Integration

REST API with sub-200ms latency; mobile and web apps

Scalability

Handles 1000 requests per second efficiently





Conclusion and Future Work

Current Achievement

High-accuracy system reducing crop loss by 15%

Future Enhancements

- Expand data sources and disease categories
- Use advanced architectures like EfficientNet
- Deploy on edge devices for real-time use