Smart Sorting Project Documentation

Title: Smart Sorting – Transfer Learning for Identifying Rotten Fruits and Vegetables

1. INTRODUCTION

1.1 Project Overview

The Smart Sorting project aims to automate the classification of fruits and vegetables into fresh or rotten categories using deep learning. This system is useful in agricultural industries, warehouses, and food sectors to maintain food quality and reduce waste. We used transfer learning to improve accuracy with minimal training time.

1.2 Purpose

- To build a model that accurately detects fresh or rotten produce.
- To deploy the model using a web-based interface for real-time results.
- To create a reliable, user-friendly system for food quality assessment.

Team Members:

Burla Devi Vara Prasad

Bhattu Jagadeesh Prasad

Ch Vamsi

Boina Venkat

2. IDEATION PHASE

2.1 Problem Statement

Automating the sorting of fruits/vegetables to identify freshness using machine learning and computer vision.

2.2 Empathy Map Canvas

User motivations, frustrations, behaviors — can be described briefly or in a table.

2.3 Brainstorming

Initial ideas included image-based classification, hardware integration (conveyor belt), and multi-fruit detection. We selected CNN-based transfer learning for simplicity and accuracy.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User uploads image → Backend processes → System outputs class (fresh/rotten).

3.2 Solution Requirement

- Dataset of fruits/vegetables
- Model training in Google Colab
- Flask app deployment
- Frontend to upload and display predictions

3.3 Data Flow Diagram

Image Upload → Preprocessing → CNN Model → Prediction

3.4 Technology Stack

- Python, TensorFlow/Keras
- Google Colab
- Flask
- HTML/CSS frontend
- Google Drive for dataset storage

4. PROJECT DESIGN

4.1 Problem-Solution Fit

Manual sorting is time-consuming and error-prone. A CNN model improves speed and accuracy.

4.2 Proposed Solution

Use transfer learning with a pre-trained model (like MobileNetV2) and fine-tune it for fruit classification.

4.3 Solution Architecture

Block diagram showing image input → model → output (visual can be added).

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

- Week 1: Dataset preparation
- Week 2: Model building
- Week 3: Web development (Flask + HTML)
- Week 4: Testing and deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Model Accuracy: ~95%
- Correct Prediction Mapping ensured using saved `class_indices.json`
- Flask App: Successfully predicts uploaded image class with correct label

7. RESULTS

7.1 Output Screenshots

- Displayed class names (e.g., “rottenbanana”) as output.
- Image upload UI for prediction
- Screenshot of classified results

8. ADVANTAGES & DISADVANTAGES

Advantages

- Fast and accurate detection
- Easy web interface
- Minimal training required using transfer learning

Disadvantages

- Limited classes (only 16 fruits/veggies)
- Needs GPU for faster training

9. CONCLUSION

The Smart Sorting system combines deep learning and web technologies to provide a simple yet powerful tool for food quality analysis. It automates a critical part of the food industry and opens doors for advanced smart farming applications.

10. FUTURE SCOPE

- Add more classes
- Raspberry Pi-based real-time deployment
- Conveyor-based automation
- Mobile app interface

11. APPENDIX

Source Code

`fruit_classifier.h5`, `class_indices.json`

Dataset Link

https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification

GitHub & Project Demo Link

GitHub Repository:

Demo video Link: https://drive.google.com/file/d/17nigjZd6itH9wQUnlEfExII-wVO_ms5N/view?usp=drive_link