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 is designed to classify fruits and vegetables as either fresh or rotten using a deep learning approach. This application is especially beneficial for agricultural industries, food storage units, and retail environments aiming to reduce waste and improve produce quality. Leveraging transfer learning significantly reduces training time while achieving high accuracy.

1.2 Purpose

- Develop an efficient model that detects the freshness of fruits and vegetables.
- Implement a web interface to interact with the model in real time.
- Provide a user-friendly platform for food quality analysis.

👥 Team Members:

- Karapakula Balaji Sai Vrushadree
- D Bhanu Parakash
- K Bhanu Prakash
- Tapetta Chinna Sanjamma

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

- High prediction accuracy
- Easy-to-use web interface
- Rapid development due to transfer learning

Disadvantages

- Dataset limited to 16 produce classes
- Model training requires significant compute (GPU recommended)

9. CONCLUSION

Smart Sorting demonstrates how transfer learning and web technologies can be combined to create an impactful solution for food quality analysis. It helps reduce human error, improve accuracy, and has potential real-world applications in smart farming and supply chain automation.

10. FUTURE SCOPE

- Expand classification to additional fruit and vegetable types
- Edge deployment using Raspberry Pi
- Conveyor belt integration for real time detection
- Mobile application version

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: https://github.com/Vrushadree10/smart-sorting-transfer-learning-for-identifying-rotten-fruits-and-Vegetables/blob/main/Smart%20Sorting%20Project.ppt

Demo video Link: https://drive.google.com/file/d/1oODqPoJGiblFpisEfeei71PFMcfKE-Dj/view?usp=sharing

-