

Smart Sorting Project Documentation

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

1. INTRODUCTION

1.1 Project Overview

Smart Sorting is an AI-based system that uses transfer learning to detect rotten fruits and vegetables from images. By fine-tuning pre-trained deep learning models on a custom dataset, it achieves high accuracy in classifying fresh vs. rotten produce. The system is deployed through a Flask web app for real-time image prediction. It automates quality checks in food processing plants, supermarkets, and smart homes, helping reduce manual labor and food waste. This project showcases the practical use of deep learning in agriculture and food safety.

1.2 Purpose

- Minimize food waste by identifying spoiled items early in the supply chain or at home.
- Ensure customer satisfaction by delivering only fresh, high-quality fruits and vegetables.
- Reduce manual labor in sorting processes.
- Improve accuracy and speed of quality control using deep learning-based image classification.

Team Members:

- Tarun kumar
- Nangedda Leela Shakunya Devi
- Nangedda Baby Sai Nandhini
- Narra Simhadri

2. IDEATION PHASE

2.1 Problem Statement

Identifying rotten fruits and vegetables manually is slow, inconsistent, and prone to errors. This results in food waste and inefficiency in quality control. An automated solution is needed to improve accuracy and speed using deep learning.

2.2 Empathy Map Canvas

- Improve work efficiency and reduce effort
- Maintain quality and customer satisfaction

2.3 Brainstorming

Initial ideas included image-based classification, conveyor belt integration, and multi-fruit detection. Traditional methods were considered but lacked accuracy. We chose CNN-based transfer learning for its simplicity and high performance.

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

- Accurate detection
- Faster sorting
- Less manual work
- Reduced food waste
- Consistent quality

Disadvantages

- Requires good-quality images
- Needs proper lighting conditions
- Initial setup can be costly
- Limited accuracy on unfamiliar items

9. CONCLUSION

The Smart Sorting system combines deep learning and web technologies to provide 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 fruit and vegetable classes for broader detection
- Raspberry Pi-based real-time deployment for edge computing
- Conveyor-based automation for industrial sorting
- Mobile app interface for easy user interaction

11. APPENDIX

Source Code

`app.py`, `index.html`

Dataset Link

<https://www.kaggle.com/code/osamaabobakr/fruit-and-vegetable-disease-healthy-vs-rotten/input>

GitHub & Project Demo Link

GitHub Repository : <https://github.com/leela2005/Smart-Sorting-Transfer-Learning-for-Identifying-Rotten-Fruits-and-Vegetables>

Demo video Link:

<https://drive.google.com/file/d/1z4a3EwAg7z9CnA3E6bKiEufiSkPNx89A/view?usp=sharing>

-