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

Team Information

- Team ID: LTVIP2025TMID34085
- Team Size: 4 members
- Team Leader: Venkateswari Chimata
- Team Members: Venkateswari Chimata,Vankayalapati Nikhita,Veeranki Siva Lakshmi,Vangapandu Rakesh

Table of Contents

---

1. Project Overview:

Smart Sorting is an innovative project using **transfer learning** to accurately identify rotten fruits and vegetables, significantly enhancing **sorting efficiency** and **quality control** in various sectors. By fine-tuning pre-trained deep learning models on specific produce datasets, the system can automate the detection of spoilage in food processing plants, ensuring only fresh produce reaches consumers in supermarkets, and even help households reduce food waste by monitoring refrigerator contents and alerting users to items nearing spoilage. This adaptable technology promises to reduce waste, improve product quality, and streamline operations across the agricultural and food industries, making it a valuable tool for a more sustainable food ecosystem.

---

2. System Architecture

The system follows a modular architecture with three main layers:

a. Frontend Layer

- HTML/CSS templates to enable user interaction.
- File input to upload fruit/vegetable images.

b. Backend Layer (Flask)

- Handles routing, image uploads, and rendering templates.
- Passes image data to the model for prediction.

c. Model Layer

- A Convolutional Neural Network (CNN) trained using transfer learning (e.g., MobileNet).
- Outputis weather the given fruit/vegetable is rotten or not based on image analysis.

Workflow:

1. User uploads an image via the frontend.
2. Flask receives the request and saves the image.
3. Image is preprocessed and fed into the model.
4. Model returns prediction.
5. Flask renders the result on the web page.

---

3. Technology Stack

- Language: Python
- Framework: Flask
- Deep Learning: TensorFlow, Keras
- Frontend: HTML5, CSS3
- Libraries:
    - numpy (numerical operations)
    - Pillow (image processing)
    - tensorflow.keras (model training and inference)

---

4. Project Structure

Rotten fruits or vegetables classifier-app/

```
├── app.py                          # Main Flask app
├── fruit_freshness_mobilenetv2.h5  # Trained CNN model
├── static/
│   └── uploads
├── templates/
│   ├── index.html
```

---

5. Implementation Details

- The model is trained on a dataset of Rotten and fresh fruits or vegetables labeled images.
- Images are resized to 224x224 and normalized.
- A MobileNet-based CNN is used for better generalization and fast predictions.
- The model is saved as fruits .pkl after training.
- Flask loads the model at runtime and performs inference on uploaded images.

---

6. Development Workflow

1. Data collection and preprocessing.
2. Model design and training using Keras in a Jupyter notebook.
3. Export trained model.
4. Build Flask app (app.py) to serve the model.
5. Create HTML templates for the frontend.
6. Link form submission to prediction logic.
7. Test locally and refine the user experience.

---

7. Setup and Installation

Requirements:

- Python 3.8+
- Pip install tensorflow

Installation Steps:

Step 1: Navigate to project director

cd fruits_veg_app

Step 2: Install dependencies

pip install -r requirements.txt

Step 3: Run the application

python app.py

Access:

Open your browser and go to: http://127.0.0.1:5000/

---

8. Features and Functionality

- Clean and intuitive user interface.
- Upload fruits or vegetables images in .jpg or .png format.
- Real-time predictions using a trained deep learning model.
- Display of predicted type along with uploaded image.
- Error handling for invalid uploads or unsupported files.

---

9. API Documentation

GET /

- Loads the homepage.
- Returns the upload form.

POST /predict

- Accepts an image file.
- Preprocesses the image and predicts rotten or fresh type.
- Returns the result page with the prediction weather rotten or fresh.

---

10. Screenshots and Results

Home Page:

- File input and upload button.

Result Page:

- Displays uploaded image.
- Shows predicted result (e.g., "Rotten").

Model Accuracy:

- Validation accuracy: ~95% (can be updated based on final model).

---

11. Challenges and Solutions

Challenge	Solution
Similar appearance across classes	Used transfer learning for robust feature learning
Large model size	Optimized with MobileNet to reduce size
UI responsiveness	Applied lightweight custom CSS
Handling unsupported file formats	Added file validation in Flask

---

12. Future Enhancements

- Deploy on cloud (Render, AWS, or Heroku).
- Add mobile responsiveness to UI.
- Use cloud storage for uploaded images.
- Implement top-3 prediction output.
- Add multilingual support.
- Train on larger and more diverse fruits or vegetables datasets

---

13. Conclusion

The "Smart Sorting" project, leveraging **transfer learning** to identify rotten fruits and vegetables, offers a powerful solution to **reduce food waste** and enhance **quality control** across the entire food supply chain. By efficiently automating spoilage detection in settings ranging from large food processing plants and supermarket receiving docks to smart home refrigerators, this initiative not only improves operational efficiency and ensures fresher produce for consumers but also significantly contributes to a more sustainable and economically viable food ecosystem. This practical application of AI underscores its vital role in creating smarter, more efficient, and sustainable food management practices globally.

Git Hub Link : https://github.com/nikhitavankayalapati/smart_sorting/tree/main