SMART-SORTING: TRANSFER LEARNING FOR IDENTIFYING ROTTEN FRUITS AND VEGETABLES

Problem Statement

Food waste is one of the major challenges facing the agricultural and retail industries worldwide. According to various studies, nearly one-third of all food produced globally is lost or wasted. Fruits and vegetables, being perishable items, account for a significant share of this wastage.

One critical factor contributing to this problem is the difficulty of detecting and sorting rotten produce during storage, transportation, and at the point of sale. Manual inspection is the most common method currently used to identify spoilage. However, it is slow, labor-intensive, subjective, and prone to human error.

Workers may overlook subtle signs of spoilage or misclassify good produce as rotten, leading to financial losses, customer dissatisfaction, and unnecessary waste. Additionally, in large-scale facilities such as warehouses, markets, or supermarkets, inspecting thousands of items daily is neither scalable nor cost-effective.

There is a strong need for an **automated, scalable, and reliable system** to quickly and accurately detect rotten fruits and vegetables. Such a solution can help reduce food waste, improve supply chain efficiency, enhance customer satisfaction, and contribute to sustainability goals.

Proposed Solution

To address these challenges, this project proposes a **machine learning–based solution** that leverages deep learning and transfer learning techniques to identify whether fruits and vegetables are fresh or rotten from images.

The core idea is to train a computer vision model to automatically learn and recognize the visual differences between fresh and rotten produce. Transfer learning enables us to use powerful pre-trained models, such as MobileNetV2 or ResNet, which have already learned rich visual features from millions of images. By fine-tuning these models on our specific dataset of fresh and rotten items, we can achieve high accuracy even with a relatively smaller custom dataset.

The key advantages of this solution are:

- **Automation:** Eliminates the need for manual inspection.
- **Speed:** Provides near-instant results, making it feasible for high-throughput sorting.
- **Accuracy:** Reduces human error by consistently applying learned features.
- **Scalability:** Can be deployed in markets, warehouses, farms, or even mobile devices for field use.
- **Cost-effectiveness:** Lowers labor costs and reduces waste-related losses.

To make this system accessible to users with no technical background, I also built a **web-based application** using Flask. This app allows users to simply upload an image of a fruit or vegetable and receive an immediate prediction indicating whether it is fresh or rotten. This ensures that the solution is practical, user-friendly, and ready for real-world deployment.

Project Workflow

The development of this project followed a structured workflow to ensure clarity, reproducibility, and effectiveness. The major stages are as follows:

✅ Data Collection and Preparation

- Collected images of various fruits and vegetables in both fresh and rotten conditions from multiple sources.
- Organized the dataset into clearly labeled categories for training.
- Applied **data augmentation** techniques such as rotation, flipping, zooming, and brightness adjustments to artificially increase dataset size and diversity, which helps improve model robustness.
- Preprocessed images by resizing them to uniform dimensions and normalizing pixel values to make them suitable for model input.

✅ Model Building with Transfer Learning

- Selected a well-known pre-trained convolutional neural network (CNN) architecture, such as MobileNetV2, which balances accuracy with computational efficiency.
- Removed the final classification layer of the pre-trained model and replaced it with custom layers tailored to our binary classification task (fresh vs. rotten).
- Fine-tuned the model on our custom dataset, allowing it to learn domain-specific features while leveraging the general visual features already learned.
- Chose appropriate hyperparameters, loss functions (e.g., binary cross-entropy), and optimizers (e.g., Adam) for effective training.

✅ Model Training and Evaluation

- Split the dataset into training, validation, and testing sets to ensure unbiased evaluation.
- Trained the model while monitoring performance metrics such as accuracy, loss, and validation accuracy.
- Used techniques such as early stopping and learning rate scheduling to prevent overfitting.
- Evaluated the trained model on the validation and test sets, analyzing metrics and confusion matrices to assess its ability to generalize to unseen data.

✅ Prediction and Testing

- Ran tests with new images to verify that the model correctly predicts whether the item is fresh or rotten.
- Documented the model's accuracy and limitations.
- Conducted error analysis to identify any systematic misclassifications.

✅ Web Application Development and Deployment

- Developed a simple yet functional web application using Flask.
- Created an intuitive HTML front-end where users can upload images directly from their device.
- Integrated the trained model into the Flask back-end to handle image preprocessing and prediction.
- Ensured that the result is clearly displayed to the user, indicating whether the uploaded image shows fresh or rotten produce.
- Designed the app to be lightweight and easy to deploy locally or on cloud servers for broader access.

✅ Advantages of the Workflow

- Modular design allows for easy updates or improvements.
- Code reusability and clean structure support scalability.
- User-friendly interface ensures accessibility for non-technical users.
- Potential for integration with larger supply-chain systems or mobile applications.