Artificial Intelligence and Machine Learning
Project Documentation

1. **Introduction**
   - **Project Title:** Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables
   - **Team Members:** Team Leader : Jami Yamini

     Team member : Ganta Bhargavi Rani

     Team member : Geethika Ede

     Team member : Gopireddy Pritham Reddy

2. **Project Overview**
   - **Purpose:** The primary purpose of this project is to develop an AI-driven solution that helps to consumers, vendors to accurately determine the freshness of fruit and vegetables and to provide fast and accurate result.
   - **Features:**
     Automatically classifies fruits and vegetables as fresh or rotten

     Uses VGG16 transfer learning model for accurate predictions

     Supports 28 categories of fruits and vegetables from the Kaggle dataset

     Flask web application for easy access

     Simple image upload functionality

     User-friendly interface built with HTML, CSS, and Bootstrap

     Real-time prediction display

3. **Architecture**
   - **Frontend**: Web interface (HTML/CSS/Bootstrap)
   - **Backend**: Flask API server
   - **AI Model**: VGG16 transfer learning model
   - **Data Flow**: User → Image Upload → Flask API → VGG16 → Prediction → Display Output
4. **Setup Instructions**
   - **Prerequisites:** Python 3.7 or higher

     Modern web browser with JavaScript enabled

   - **Installation:**

     1. Clone the repository from GitHub

2. Create and activate a virtual environment

3. Install required dependencies: Tensorflow, Numpy, Pandas, Matplotlib, opencv-python, scikit-learn

4. Download dataset from Kaggle and extract to `data/` directory

5. Set up environment variables if needed

6. Ensure trained model file is in the `model/` directory

5. Folder Structure

Templates: Contains HTML templates for the web interface

`index.html` - Main upload and prediction page

Static: Static files

`uploads/` - Temporary storage for uploaded images

`images/` - consist images

Model: Machine learning model files

`Model file( .h5)` - Trained VGG16 model

6. Running the Application

Application Startup:

- `python app.py`

Access: Open web browser and navigate to `http://localhost:5000`

Usage Flow:

1. Navigate to the web interface
2. Upload an image of fruit or vegetable
3. Click submit/predict button
4. View the classification result (Fresh/Rotten)

7. API Documentation

POST /predict

Method: POST

Parameters: `image` (file): Image file of fruit/vegetable

Response: JSON object containing prediction result

8. Authentication

- **Current Implementation:** No authentication required for the current version

9. User Interface

Main Interface: Clean, responsive web interface with image upload functionality

Upload Section: Drag-and-drop or click-to-upload image input

Results Display: Clear presentation of prediction results with confidence scores

Responsive Design: Works across desktop and mobile devices using Bootstrap framework

10. Testing

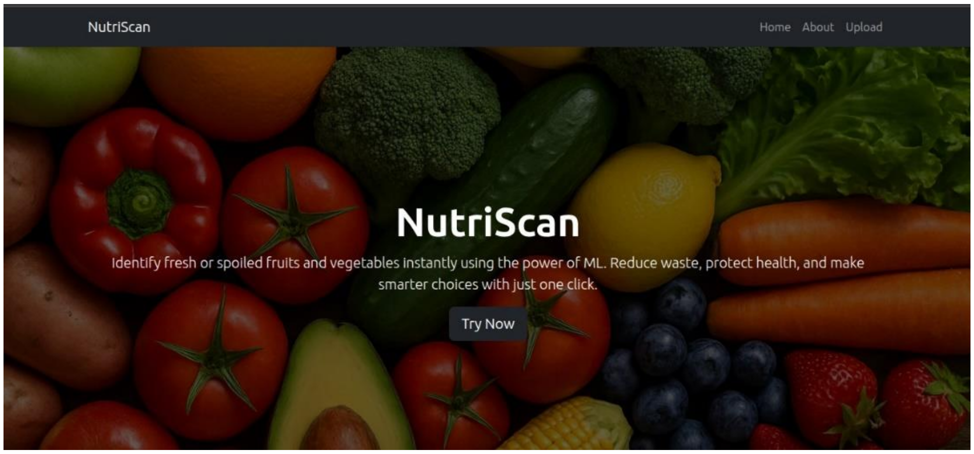
Performance testing achieved 80% validation accuracy on unseen data Manual testing with various fruit and vegetable images

Cross-validation on different image qualities and lighting conditions

11. Demo

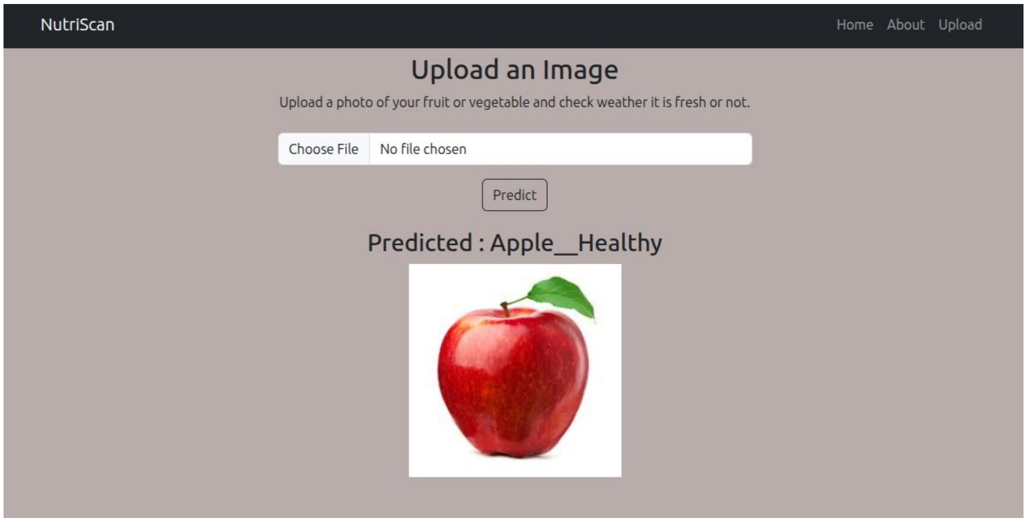
https://drive.google.com/file/d/1p3CN4RkZk3w9RbUt6A7-49EqwQ3EZlXA/view?usp=sharing

Screenshorts:



NutriScan

Identify fresh or spoiled fruits and vegetables instantly using the power of ML. Reduce waste, protect health, and make smarter choices with just one click.



Upload an Image

Upload a photo of your fruit or vegetable and check weather it is fresh or not.

Predicted : Apple__Healthy

12. Known Issues

- **Limited Categories:** Currently supports only 28 specific categories of fruits and vegetables
- **Single Image Processing:** Currently processes one image at a time

13. Future Enhancements

- **Accuracy Improvement:** Fine-tune the VGG16 model or implement ensemble methods to achieve 90%+ accuracy for more reliable commercial use
- **Multi-image Analysis:** Enable batch processing to analyze multiple fruits/vegetables simultaneously, improving efficiency for large-scale operations
- **Freshness Scoring:** Instead of binary classification, provide a freshness score (0-100%) with estimated shelf life predictions