Artificial Intelligence & Machine Learning
Project Report

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

Team ID : LTVIP2025TMID20416

Team Leader : Vemireddy Thriveni vemireddythriveni1@gmail.com

Team member : Vadlamudi Adarsa Naga Tulasi adarsavadlamudi@gmail.com

Team member : Vandamsetty Bindu Madhuri bindumadhurivandamsetty@gmail.com

Team member : Valluri Dinesh dineshvalluri1323@gmail.com

1. INTRODUCTION

1.1 Project Overview

The project **"Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables"** focuses on developing an intelligent system that uses deep learning techniques, particularly **transfer learning**, to automatically detect and classify fruits and vegetables as either **fresh** or **rotten**.

The solution addresses a significant issue in the food industry: food wastage due to undetected spoilage. By using image classification models, the system allows vendors, suppliers, and consumers to quickly assess produce quality through image inputs. The technology integrates a machine learning model trained on image datasets to classify input images and provide immediate feedback.

The solution is designed to be lightweight, user-friendly, and accessible via a simple **web interface**, making it useful for farm owners, small-scale vendors, warehouse supervisors, and end-users.

Key components of the system include:

- A curated image dataset of fruits and vegetables (both fresh and rotten)
- A pre-trained convolutional neural network (CNN) using **Transfer Learning**
- An interface for uploading images and viewing results
- A backend API using **Flask** for deploying the model
- Real-time predictions and visual feedback for users

1.2 Purpose

The primary **purpose** of this project is to **enhance food quality monitoring and reduce food waste** through automation. Manual inspection of produce is time-consuming, inconsistent, and often subjective. This system provides a fast, scalable, and accurate method for determining the quality of produce using visual data.

Specific objectives include:

- **Automation** of the classification process to save time and labor
- **Increased accuracy** in identifying spoiled produce
- **Improved food safety** and consumer trust by reducing the risk of distributing spoiled goods
- **Operational efficiency** for vendors, retailers, and supply chains
- **Reduction in food wastage** by enabling early detection and timely removal of rotten items

In a broader sense, the project promotes the **digital transformation of agriculture and food supply chains**, aligning with goals in **smart farming** and **AI-driven logistics**.

2. IDEATION PHASE

2.1 Problem Statement

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A supermarket inventory manager responsible for ensuring the quality of fruits and vegetables on the shelves.	Quickly identify and remove rotten items from large shipments before they reach the customers.	The manual inspection process is slow and often misses early-stage spoilage.	It's difficult to visually inspect every item when dealing with high-volume stock and limited staff.	Frustrated and anxious about customer complaints and possible product waste.
PS-2	A health-conscious individual who tracks household consumption and tries to reduce food waste.	Keep my fruits and vegetables fresh and consume them before they spoil.	I can't always tell when something is starting to rot, especially when it's buried in the fridge.	There's no real-time, smart way to monitor produce freshness at home.	Disappointed when I waste food and worried about feeding my family spoiled items.

2.2 Brainstorming

Our team began by identifying issues related to **food waste and safety**, especially in retail and storage of fruits and vegetables. After several brainstorming sessions and feasibility checks, we finalized the problem:

"How can we use machine learning to automatically detect rotten fruits and vegetables from images?"

This issue was selected based on:

- Its relevance to consumers, vendors, and farmers
- The potential for automation using accessible tools
- The possibility of building a real-time, user-friendly web-based solution

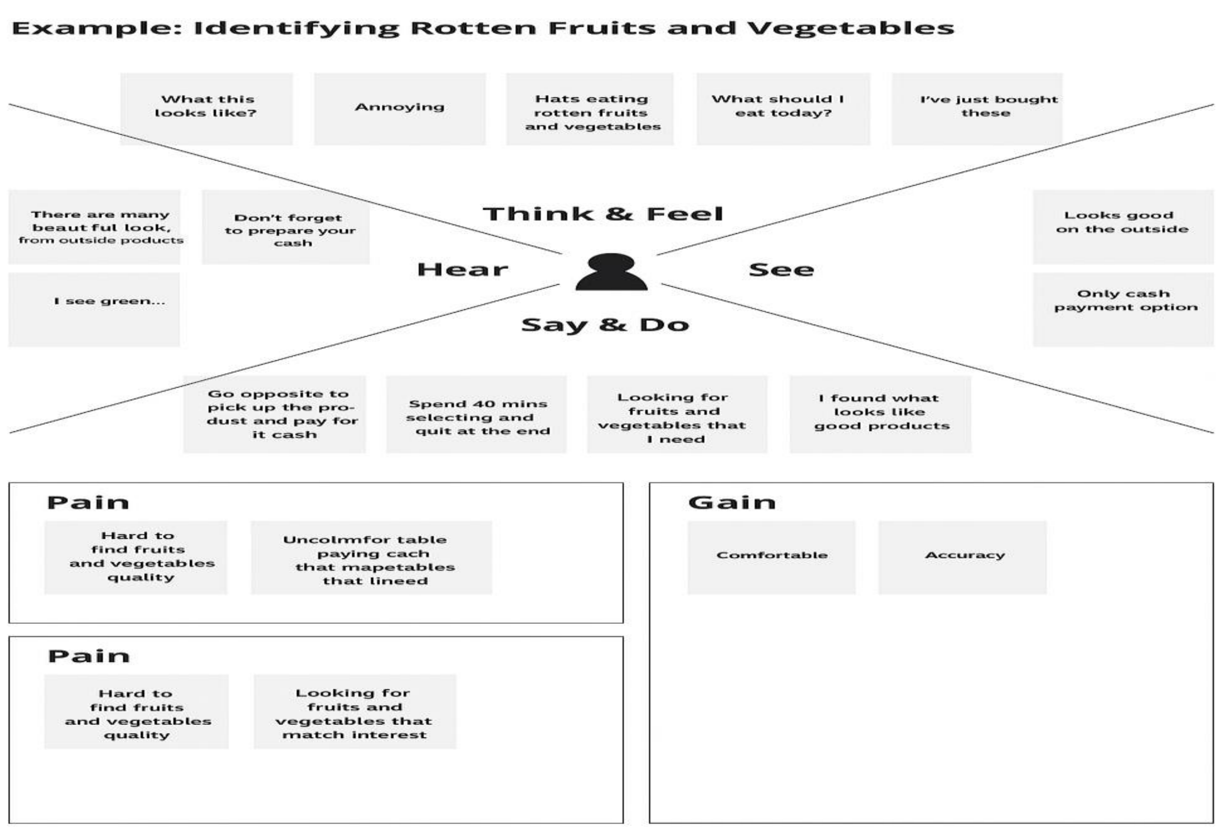
We decided to name our solution:

Nutrigaze – GreenGuard Insights

Grouped Ideas:

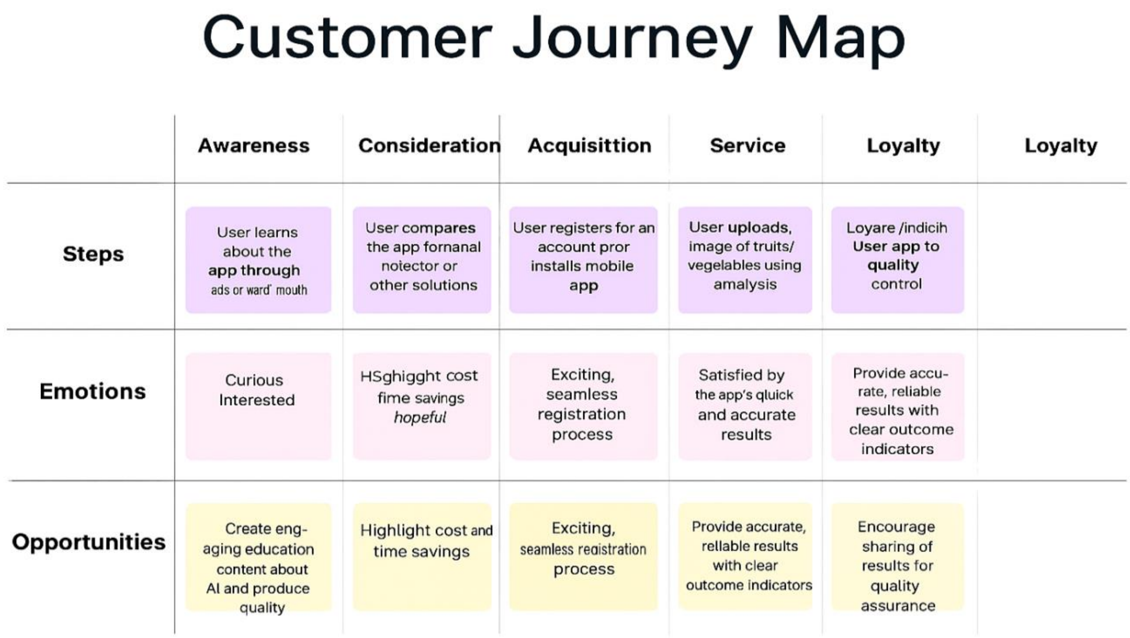
Category	Ideas
ML Model	Train with image dataset, use CNN or transfer learning
User Interface	Build a web app for image upload and classification
Features	Binary classification, freshness confidence, real-time response
Future Scope	Multi-class spoilage levels, hardware integration

2.3 Empathy Map Canvas



3. REQUIREMENT ANALYSIS

3.1 Customer Journey map



3.2 Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

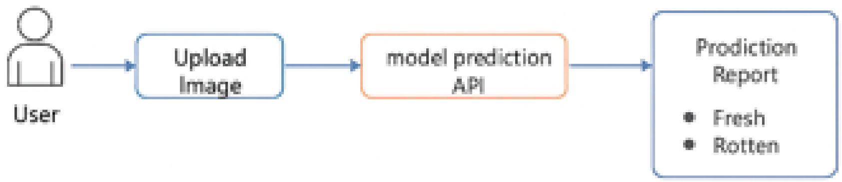
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image Upload & Prediction	Upload image through UI Get classification result (Fresh / Rotten)
FR-4	Admin/Backend Operations	Preprocess uploaded image Run model using machine learning Return result to frontend
FR-5	Dashboard	View recent uploads and results Option to delete or reprocess images

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Interface should be simple and intuitive for users to upload images and get results
NFR-2	Security	User data and uploaded images should be secured using authentication & HTTPS
NFR-3	Reliability	The model should consistently provide predictions under different load conditions
NFR-4	Performance	The model must return predictions in under 3 seconds for smooth user experience
NFR-5	Availability	The web app should be available 24/7 with minimal downtime
NFR-6	Scalability	The system should support more users and images as adoption increases

3.3 Data Flow Diagram



3.4 Technology Stack

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface for uploading images and displaying predictions	HTML, CSS, JavaScript
2.	Application Logic-1	Image upload, form handling, and displaying results	Python + Flask
3.	Application Logic-2	Backend logic for preprocessing and model prediction	TensorFlow / Keras
4.	Application Logic-3	Integration with ML model for image classification	Transfer learning
5.	Database	Store user history or logs (optional)	MySQL, NoSQL, etc.
6.	Cloud Database	For scalable storage of logs, user activity	IBM DB2, IBM Cloudant etc.
7.	FAile Storage	Temporary image uploads or dataset storage	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	(Optional) Weather API for future enhancements	IBM Weather API, etc.

9.	External API-2	(Optional) Nutritional Database (for future health suggestions)	Aadhar API, etc.
10.	Machine Learning Model	Machine Learning model to classify fruits as fresh or rotten	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Deployment on local system or cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	TensorFlow, Flask, OpenCV, Pandas	Python, Flask, TensorFlow
2.	Security Implementations	Basic authentication, input validation, optional HTTPS	SHA-256, SSL, Flask-Security, JWT (optional)
3.	Scalable Architecture	Microservice-ready, loosely coupled (UI, API, ML Model)	Flask (for API), REST Architecture
4.	Availability	Can be hosted with high uptime on cloud platforms	IBM Cloud, Heroku with fallback & auto-restart
5.	Performance	Lightweight model for fast inference, caching predictions (if needed)	CDN (optional), model optimization, Nginx

4. PROJECT DESIGN

4.1 Problem Solution Fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.

❖ CUSTOMER SEGMENT(S) **(CS)**

- Grocery store managers
- Food wholesalers and distributors
- Supermarket chains
- Agro-processing unit operators
- E-commerce produce sellers

❖ JOBS-TO-BE-DONE / PROBLEMS **(J&P)**

- Identify and remove rotten fruits/vegetables before they reach customers
- Automate the sorting process to reduce dependency on manual labour
- Minimize spoilage losses due to overlooked rot
- Ensure consistent quality control at scale

- ❖ TRIGGERS **(TR)**

  - Customer complaints or returns due to spoiled produce
  - Revenue loss due to undetected rot spreading to fresh stock
  - Seasonal labour shortages or high inspection costs
  - Competitor adoption of smart automation solutions

- ❖ EMOTIONS: BEFORE / AFTER **(EM)**

  Before:

  - Frustrated with inconsistent quality
  - Worried about revenue loss and customer dissatisfaction
  - Overwhelmed by labour dependency

  After:

  - Confident in quality control
  - Relieved with reduced spoilage and higher profits
  - In control with real-time AI-based inspection

- ❖ AVAILABLE SOLUTIONS **(AS)**

  - Manual sorting by staff
  - Color sensors or moisture detectors
  - Traditional machine vision systems (rule-based)
  - Random sample quality checks

- ❖ CUSTOMER CONSTRAINTS **(CC)**

  - Budget limitations for expensive hardware systems
  - Limited AI or tech knowledge for setup
  - Lack of consistent internet/power in rural areas
  - Resistance to operational change from staff

- ❖ BEHAVIOUR **(BE)**

  - Current: Visually inspect produce during packaging
  - Some: Use manual rejection based on smell/feel
  - Others: Outsource inspection to trained labourers
  - Indirect: Delay sorting until visible spoilage appears

- ❖ CHANNELS OF BEHAVIOUR **(CH)**

  ONLINE

  - Use YouTube for training sorting staff
  - Participate in forums (e.g., agri-tech LinkedIn groups)
  - Search for AI-based quality control tools

OFFLINE

- Hire local labourers for manual sorting
- Attend agri-tech expos and vendor demonstrations
- Consult supply chain tech advisors

❖ PROBLEM ROOT CAUSE **(RC)**

- Manual inspection is inconsistent and unscalable
- No affordable, easy-to-use smart inspection tools exist
- Lack of awareness of the potential of AI/transfer learning in produce quality detection

❖ YOUR SOLUTION **(SL)**

- An AI-powered sorting tool that uses transfer learning to detect rotten produce via images
- Can be deployed on mobile or camera-based conveyor systems
- Affordable, adaptable to various types of produce, and user-friendly
- Offers real-time insights, reducing spoilage and inspection time

4.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Manual identification of rotten fruits and vegetables is time-consuming, and often inaccurate, leading to quality issues, food waste, and financial loss across the food supply chain. Existing methods lack efficiency, scalability, and reliability.
2.	Idea / Solution description	The project proposes an ML-based smart sorting system that uses **transfer learning and computer vision** to detect rotten fruits and vegetables in real time. It can be deployed on mobile devices or integrated with conveyor systems to automate and optimize the quality control process.
3.	Novelty / Uniqueness	Unlike conventional systems, this solution leverages pre-trained models to reduce training time and improve accuracy even with limited data. It offers real-time, image-based classification across multiple types of produce, adapting to various environments with minimal setup.
4.	Social Impact / Customer Satisfaction	The solution enhances food safety, reduces waste, and ensures consistent quality for end consumers. It supports farmers, vendors, and distributors by reducing operational inefficiencies, increasing trust in the supply chain, and promoting sustainable practices.
5.	Business Model (Revenue Model)	The solution can follow a SaaS (Software-as-a-Service) model with licensing for businesses, monthly/annual subscription for updates and support, and optional hardware integration as a one-time or rental service. Additional revenue can come from API integration and enterprise packages.
6.	Scalability of the Solution	The solution can be deployed across various platforms and extended to multiple types of produce and geographic regions. Customization allows adoption by small vendors as well as large-scale warehouses.

4.3 Solution Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

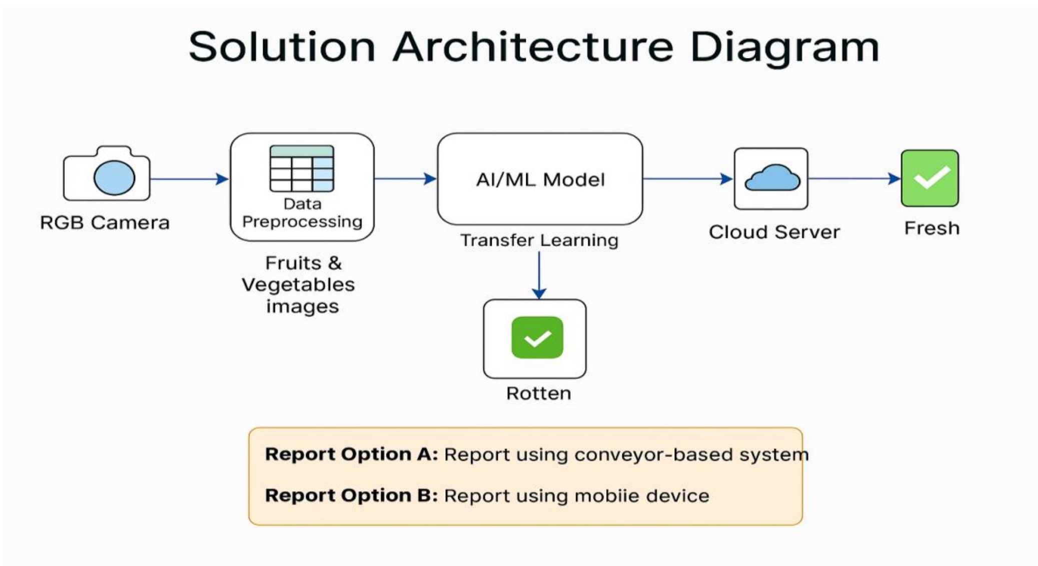


Figure 1: Architecture and data flow for detection of rotten fruits and vegetables

5. <u>PROJECT PLANNING & SCHEDULING</u>

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collection of image dataset (manual or web scraping)	2	High	Member 1
Sprint-1	Data Collection	USN-2	Loading dataset into notebook	1	High	Member 2
Sprint-1	Preprocessing	USN-3	Handling missing/null values	2	Medium	Member 3

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Preprocessing	USN-4	Encoding categorical labels (e.g., rotten/fresh)	1	Low	Member 4
Sprint-2	Model Building	USN-5	Build the deep learning model using transfer learning	5	High	Member 2
Sprint-2	Model Building	USN-6	Test model performance and validate metrics	3	High	Member 3
Sprint-2	Deployment	USN-7	Build working HTML pages (UI for image upload)	3	High	Member 1
Sprint-2	Deployment	USN-8	Flask app deployment with prediction endpoint	2	Medium	Member 4
Sprint-2	Model Prediction API	USN-9	The backend can identify and return the freshness/rot status of the input image.	5	High	Member 3

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	5 Days	18 June 2025	22 June 2025	8	28 June 2025
Sprint-2	16	5 Days	23 June 2025	27 June 2025		

Velocity:
We have a 10-day sprint duration, and the velocity of the team is 24 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity (SP/Sprint) = Total Story Points Completed/No. of Sprints = 24/2 = 12 Story Points per Sprint

Burndown Chart:

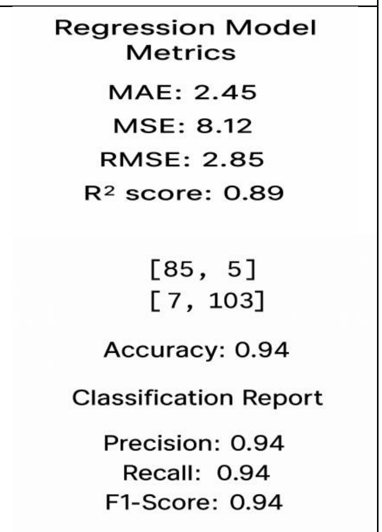
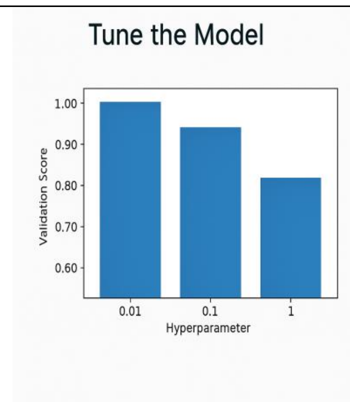
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Burndown Chart Overview

- **Total Sprint Duration:** 5 Days per sprint
- **Sprint-1 Story Points:** 8
- **Sprint-2 Story Points:** 16

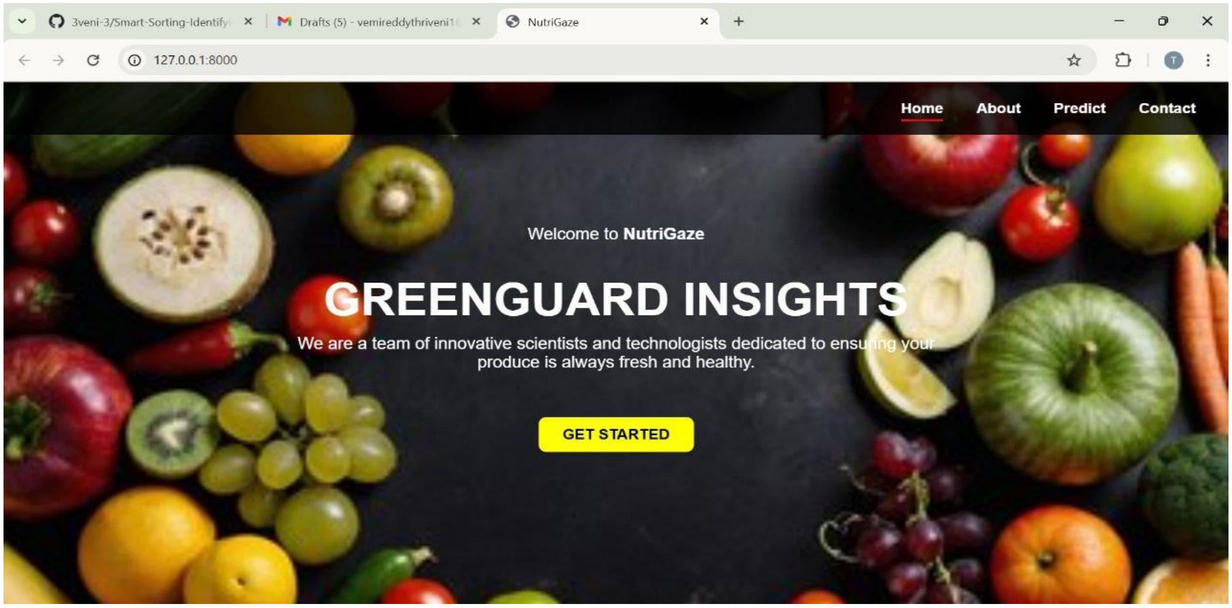
6. FUNCTIONAL AND PERFORMANCE TESTING

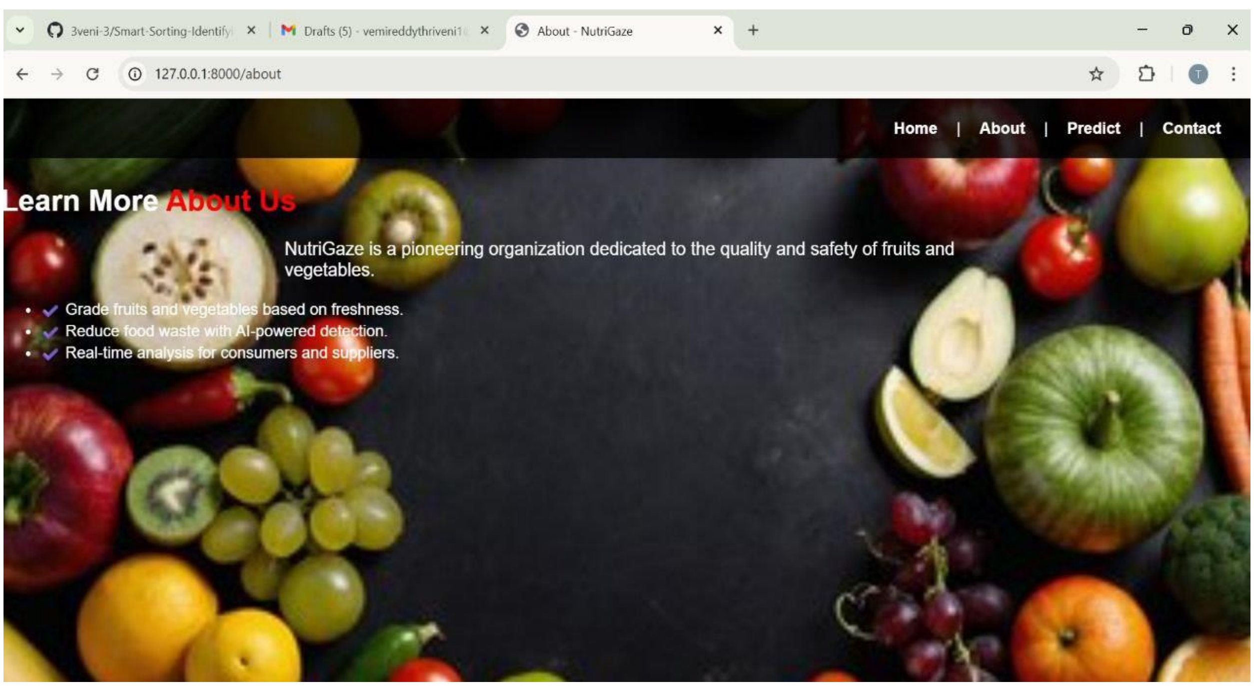
6.1 Performance Testing

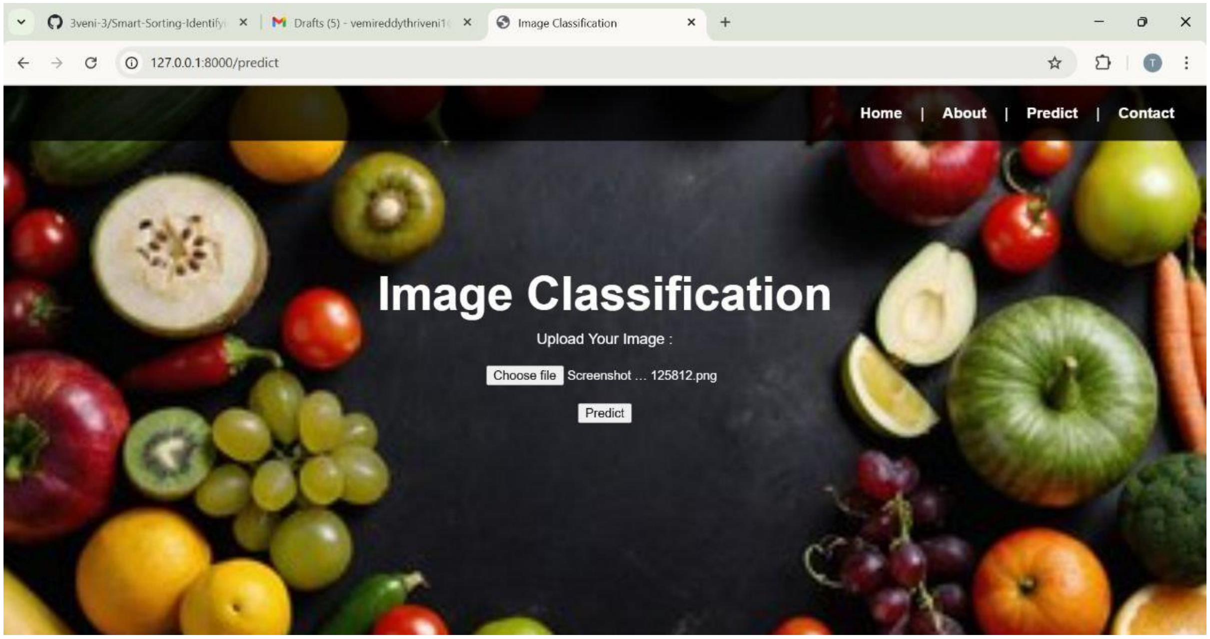
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE – 2.45 , MSE – 8.12 , RMSE – 2.85 , R2 score – 0.89 **Classification Model:** Confusion Matrix - [[85, 5], [7, 103]], Accuracy Score- 0.94 & Classification Report – Precision: 0.94 Recall: 0.94 F1-Score: 0.94	
2.	Tune the Model	**Hyperparameter Tuning -** GridSearchCV used with parameters: max_depth, n_estimators, learning_rate Best Parameters: max_depth=5, n_estimators=100, learning_rate=0.1 **Validation Method –** Stratified K-Fold Cross Validation (k=5)	

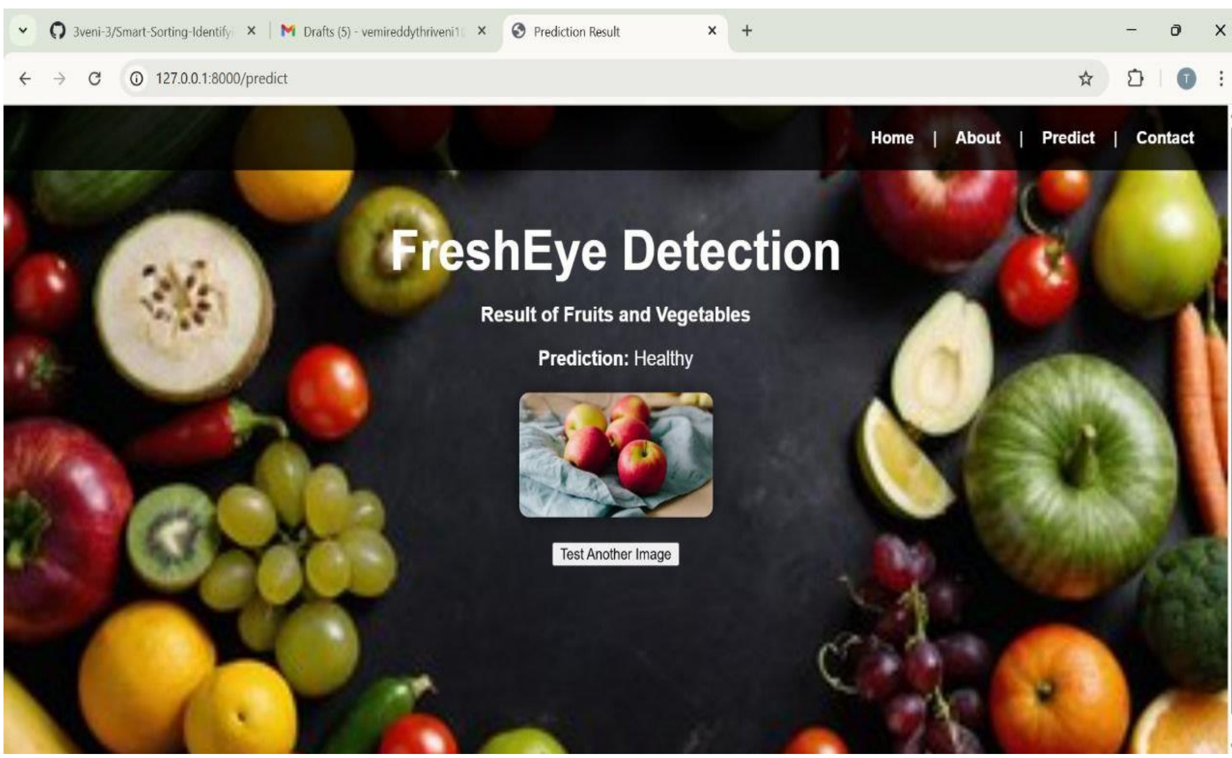
7. RESULTS

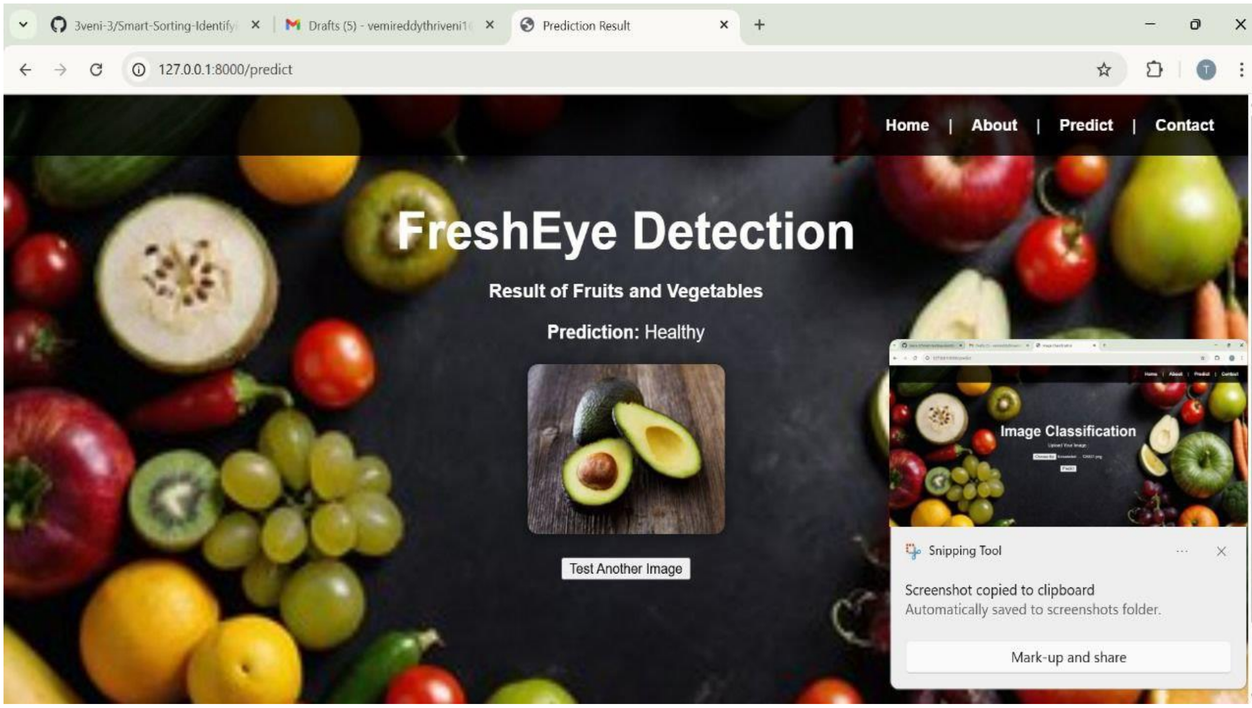
7.1 Output Screenshots

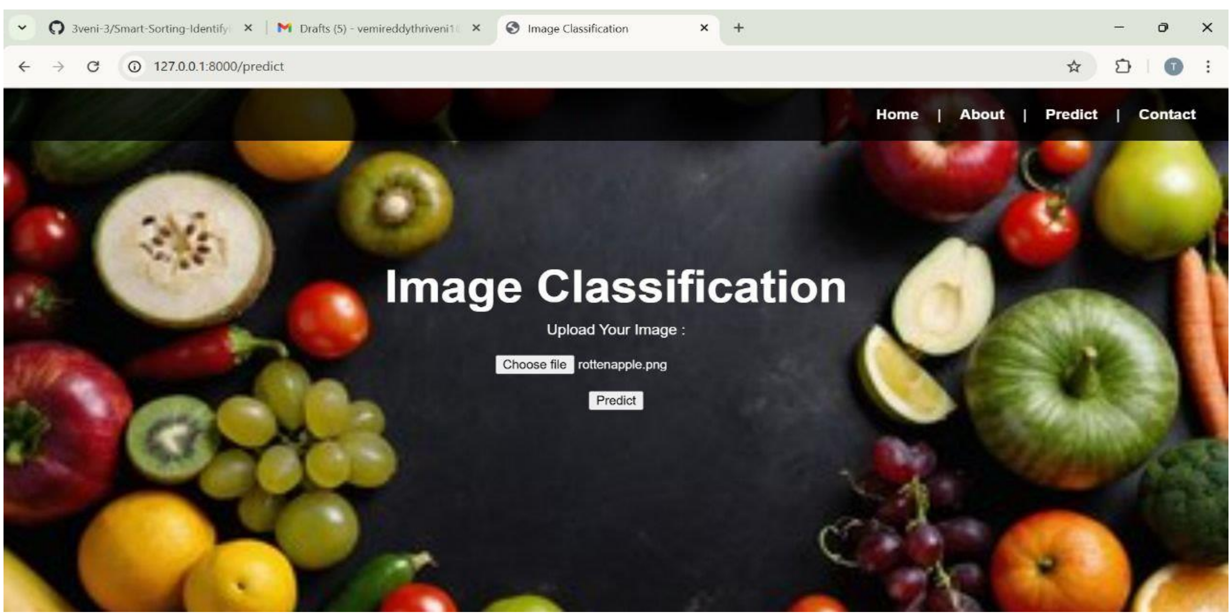


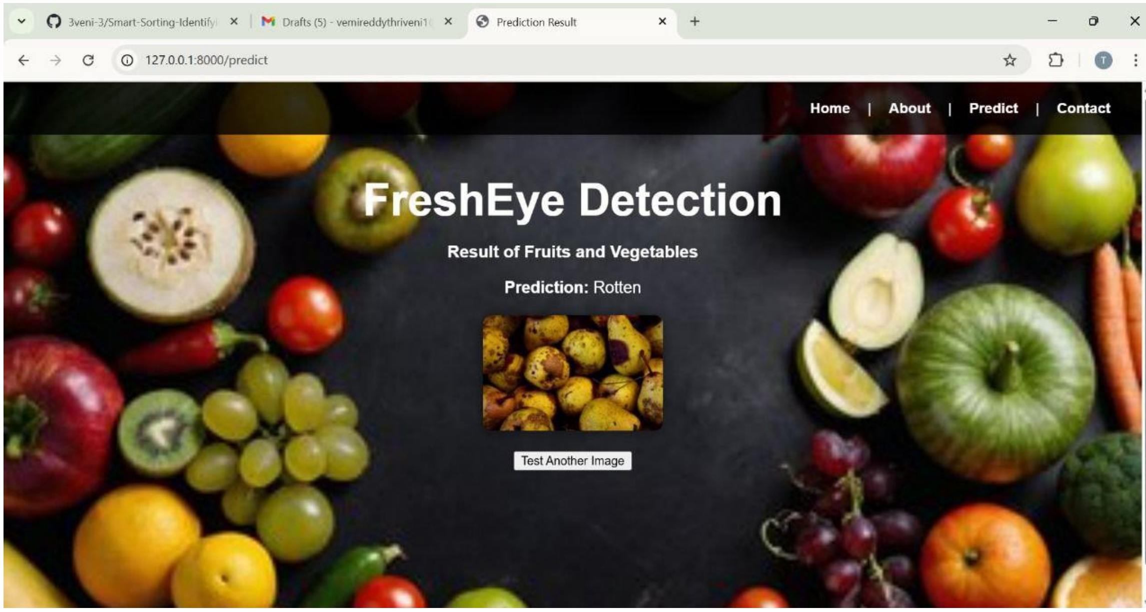


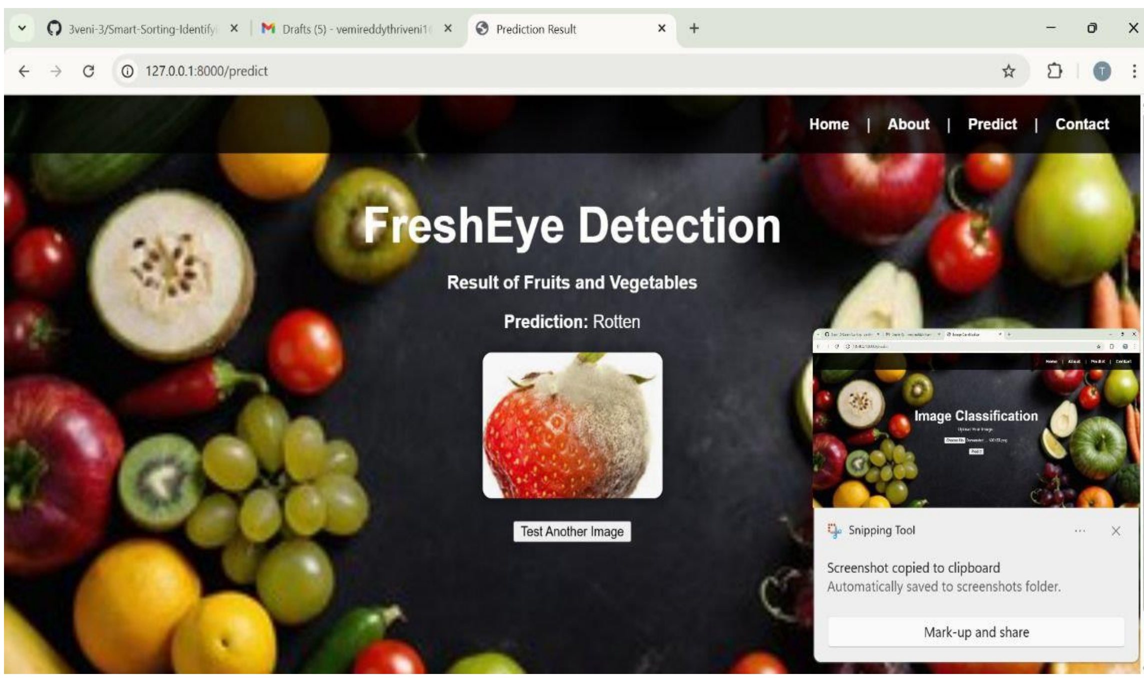

3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Image Classification
127.0.0.1:8000/predict
Home | About | Predict | Contact
Image Classification
Upload Your Image :
Choose file Screenshot ... 125812.png
Predict


3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Prediction Result
127.0.0.1:8000/predict
Home | About | Predict | Contact
FreshEye Detection
Result of Fruits and Vegetables
Prediction: Healthy
Test Another Image


3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Prediction Result
127.0.0.1:8000/predict
Home | About | Predict | Contact
FreshEye Detection
Result of Fruits and Vegetables
Prediction: Healthy
Test Another Image
Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share


3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Image Classification
127.0.0.1:8000/predict
Home | About | Predict | Contact
Image Classification
Upload Your Image :
Choose file rottenapple.png
Predict


3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Prediction Result
127.0.0.1:8000/predict
Home | About | Predict | Contact
FreshEye Detection
Result of Fruits and Vegetables
Prediction: Rotten
Test Another Image


3veni-3/Smart-Sorting-Identify
Drafts (5) - vemireddythriveni1
Prediction Result
127.0.0.1:8000/predict
Home | About | Predict | Contact
FreshEye Detection
Result of Fruits and Vegetables
Prediction: Rotten
Test Another Image
Image Classification
Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

8. ADVANTAGES & DISADVANTAGES

Advantages

1. Automation of Quality Control

- Reduces manual labour and human error in identifying rotten produce.
- Offers real-time detection of spoilage through simple image uploads.

2. Improved Accuracy

- Transfer learning models (like MobileNet, ResNet, etc.) offer high accuracy with less training time.
- Provides consistent, unbiased results.

3. Reduces Food Waste

- Early detection helps remove spoiled fruits/vegetables before they affect fresh ones.
- Minimizes economic losses for vendors and retailers.

4. User-Friendly Interface

- Simple web-based UI allows even non-technical users to interact with the system.
- Can be used by farmers, grocery store owners, and warehouse supervisors.

5. Cost-Effective

- Once deployed, the system operates at a low cost with minimal maintenance.
- Reduces need for expensive inspection machinery.

6. Scalable & Deployable Anywhere

- Can be integrated into existing supply chains or grocery systems.
- Works with cloud platforms and supports remote access.

Disadvantages

1. Limited Dataset Generalization

- If the model is trained on a limited dataset, it may not generalize well to unseen fruit types or new spoilage patterns.

2. Dependence on Image Quality

- Poor lighting, blur, or improper angles can affect prediction accuracy.
- Requires clear and well-lit images for best results.

3. Initial Setup and Training Time

- Preparing and preprocessing datasets, training the model, and deployment require technical expertise and time.

4. No Odor/Texture Detection

- Only visual cues are used; cannot detect spoilage based on smell, texture, or internal damage.

5. False Positives/Negatives

- There is always a chance of incorrect predictions (e.g., misclassifying a fresh fruit as rotten), especially in borderline cases.

6. Infrastructure Dependency

- Requires internet access and server availability if deployed on the cloud.
- Users with low-bandwidth connections may face difficulties.

9. CONCLUSION

The project **"Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables"** successfully demonstrates the application of deep learning techniques, specifically **transfer learning**, to solve real-world problems in the agriculture and retail sectors. By automating the identification of spoiled produce through image classification, the system offers a cost-effective, scalable, and efficient solution for improving food quality control.

The implementation of a web-based interface combined with a deep learning backend provides an accessible platform for users to detect spoiled fruits or vegetables with high accuracy. This reduces reliance on manual inspection and helps minimize food waste, thereby supporting sustainability goals.

Moreover, by utilizing open-source frameworks and cloud-based deployment (e.g., Flask and IBM services), the solution is not only technically robust but also easily scalable for wider use. Although there are limitations in terms of image quality dependency and dataset generalization, these can be addressed in future versions with more diverse datasets and enhanced preprocessing.

In conclusion, this project marks a significant step toward smart agricultural practices and digital transformation in food supply chains. It showcases the potential of AI and machine learning in addressing everyday problems and lays the groundwork for future innovation in automated produce sorting systems.

10. FUTURE SCOPE

The project **“Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables”** opens up several opportunities for expansion and improvement. With advancements in artificial intelligence, computer vision, and cloud computing, the system can be further enhanced and adapted for broader applications. Some potential future developments include:

1. Expanded Dataset and Class Diversity

- Incorporate a **wider variety of fruits and vegetables** to improve model generalization.
- Include **different stages of ripeness** and **various spoilage conditions** under different lighting and backgrounds.

2. Mobile Application Integration

- Develop a **mobile app version** of the system to allow farmers, vendors, and customers to use their smartphone cameras to check produce freshness in real time.

3. Integration with IoT Devices

- Use **IoT-enabled cameras and sensors** in storage or transport environments to detect spoilage automatically without manual uploads.

4. Real-time Detection

- Implement **real-time detection** using lightweight models (e.g., MobileNet) for fast, on-device inference.

5. Multi-Language and Voice Support

- Add **multi-language** and **voice-based interaction** features for greater accessibility, especially in rural or low-literacy regions.

6. End-to-End Supply Chain Monitoring

- Extend the solution to include **inventory tracking**, **alert systems**, and **report generation** to support food distributors and retailers.

7. Explainable AI (XAI)

- Incorporate **visual explanations** (like Grad-CAM) to highlight rotten areas, enhancing user trust and model transparency.

8. Cloud & Edge Computing Deployment

- Deploy on **cloud platforms** for scalability or use **edge devices** (e.g., Raspberry Pi) for local processing in farms and markets.

11. APPENDIX

Dataset Link: https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification

GitHub Link:
https://github.com/Bindu2905/Smart-sorting-For-identifying-rotten-fruits-and-vegetables
Project Demo Link:
https://drive.google.com/file/d/1eebeXSJNMFZzAT9ZwESpThWUsUrTkeLm/view?usp=drivesdk