Artificial Intelligence & Machine Learning
Project Report

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

Team ID: LTVIP2026TMIDS54490

Team Leader: K. Nanda Kishore konerunandhakishorereddy@gmail.com

Team member: M. Sai Subramanyam sainani00123@gmail.com

Team member: M. Sai Chaithanya saichaithanyamajjari@.com

Team member: C. Bhargava nanibalu0430@gmail.com

1. INTRODUCTION

1.1 Project Overview

The project titled "Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables" presents a deep learning–based image classification system that automatically detects and categorizes fruits and vegetables as either fresh or rotten. Leveraging transfer learning techniques with pre-trained convolutional neural networks, the system is designed to reduce food spoilage, improve quality assurance, and enhance operational efficiency across the food supply chain.

The proposed solution addresses a major challenge in the agriculture and retail sectors: manual inspection of perishable goods is often labor-intensive, inconsistent, and prone to error. By enabling real-time predictions from uploaded images, Smart Sorting offers an accessible, scalable, and costeffective alternative.

The system is optimized for usability and accessibility, featuring a streamlined web-based interface suitable for deployment in resource-constrained environments. Its lightweight backend enables predictions to be made swiftly, even in offline or low-connectivity conditions—making it highly applicable for small-scale vendors, farm owners, warehouse supervisors, and consumers.

Key Components:

Dataset: Curated image dataset of various fruits and vegetables in both fresh and rotten states Model Architecture: Pre-trained CNN (MobileNetV2) adapted via transfer learning for multi-class classification
User Interface: Web-based frontend for seamless image upload and visual output display
Backend Framework: Flask-based API for handling data preprocessing and model inference
Prediction Output: Instant classification with confidence score and visual result feedback

1.2 Purpose

The primary objective of this project is to enhance food quality monitoring and reduce food waste through intelligent automation. Manual inspection of fruits and vegetables is often time-consuming, inconsistent, and based on subjective judgment. This system provides a rapid, scalable, and accurate method for evaluating produce quality using visual input and machine learning.

Specific Objectives

- Automate the classification process to reduce inspection time and labor costs
- Increase accuracy in detecting early-stage spoilage in produce
- Improve food safety and enhance consumer trust by minimizing the risk of distributing degraded items
- Optimize operational efficiency for vendors, retailers, and supply chain stakeholders
- Reduce overall food wastage by enabling early detection and timely intervention

More broadly, this project supports the digital transformation of agriculture and food logistics by integrating AI into quality control workflows. It aligns with ongoing efforts toward smart farming, sustainability, and data-driven supply chain innovation.

2. IDEATION PHASE

2.1 Problem Statement

The inspection and quality assessment of fruits and vegetables in supply chains remain largely manual, subjective, and inefficient. Vendors, retailers, and consumers face consistent challenges in detecting spoilage—especially in early stages—resulting in the distribution of substandard produce, financial losses, and food waste. Traditional methods are not scalable or reliable, particularly in high-volume or low-resource environments.

There is a clear need for a fast, accurate, and accessible solution that can automate the detection of spoiled produce using minimal infrastructure, enabling better quality control and reducing unnecessary loss across the food supply chain.

2.2 Brainstorming

Our team began by identifying key challenges surrounding food safety and post-harvest losses, particularly within the retail and storage of perishable produce. Through a series of brainstorming sessions, stakeholder analysis, and technical feasibility evaluations, we consolidated our focus on a critical issue:

Problem Statement:
"How can we leverage machine learning to automatically detect and classify rotten fruits and vegetables from images?"

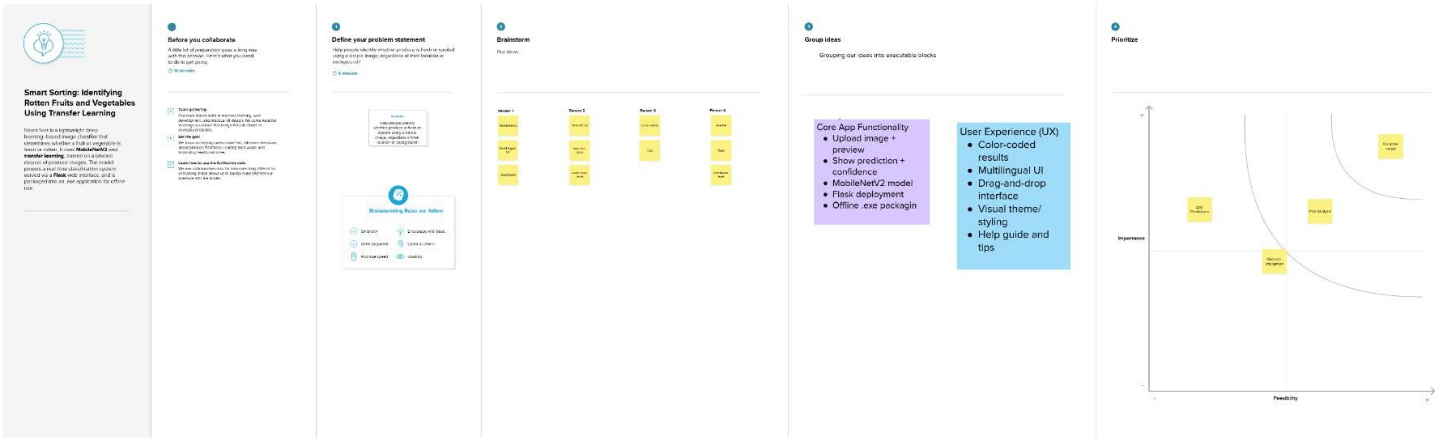
This problem was selected due to its practical relevance, real-world impact, and alignment with emerging AI-based automation trends.

Why This Problem?

- It directly impacts consumers, vendors, and farmers across the food supply chain
- It provides an opportunity to integrate automation via accessible, open-source technologies

- It offers scope for a scalable, real-time, and intuitive solution adaptable across digital platforms

Our Brainstorm Mural:
https://app.mural.co/t/smartsort1482/m/smartsort1482/1751088193260/79e439e0759f8fe62430b951277f3de2f5392f04?wid=88-1654267222937



Smart Sorting: Identifying Rotten Fruits and Vegetables Using Transfer Learning

Smart Sort is a lightweight deep learning-based image classifier that determines whether a fruit or vegetable is fresh or rotten. It uses **MobileNetV2** and **transfer learning**, trained on a labeled dataset of produce images. The model powers a real-time classification system served via a **Flask** web interface, and is packaged into an .exe application for offline use.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Define your problem statement

Help people identify whether produce is fresh or spoiled using a simple image, regardless of their location or background?

Brainstorm

Group ideas

Grouping our ideas into executable blocks

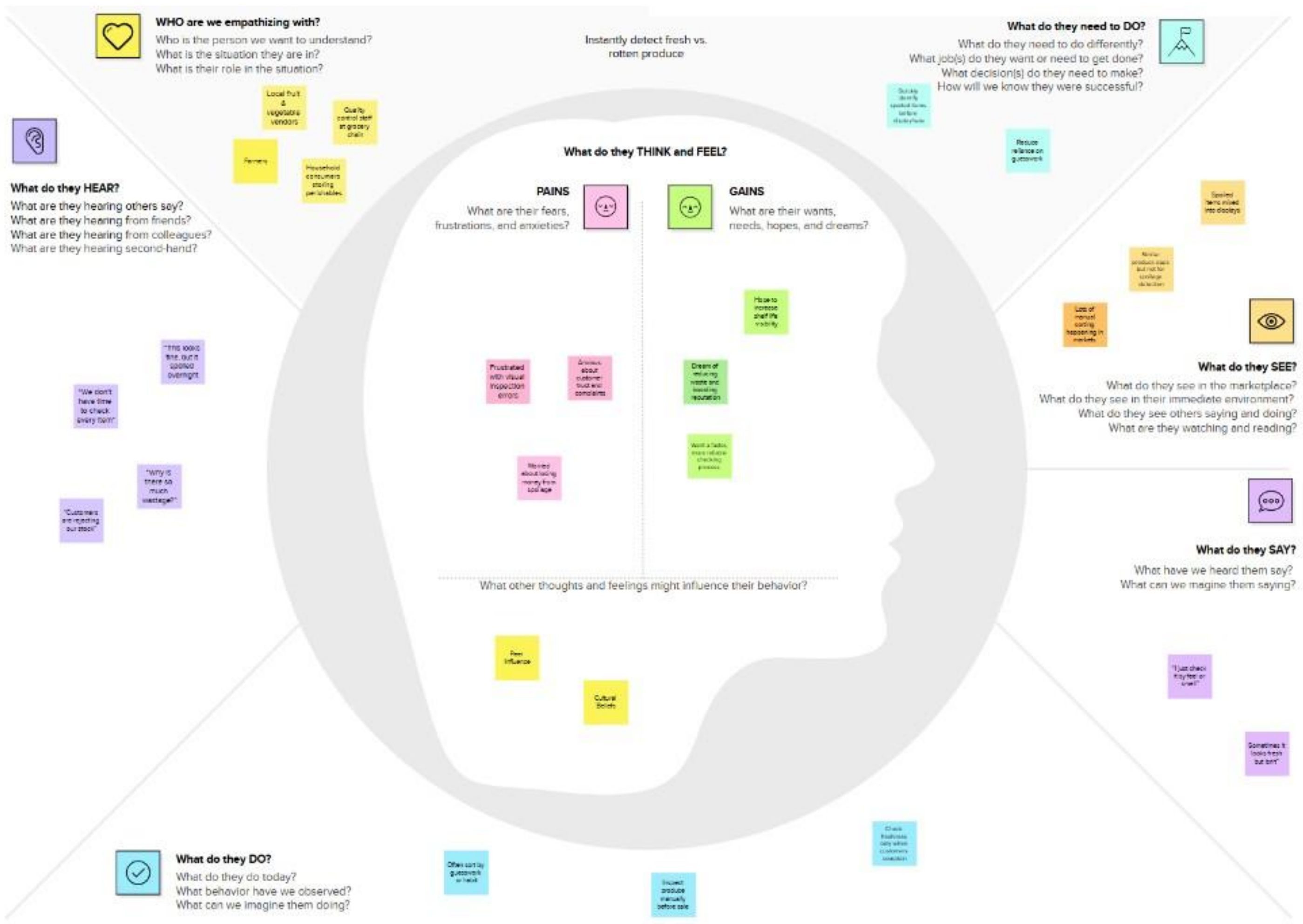
Core App Functionality
- Upload image + preview
- Show prediction + confidence
- MobileNetV2 model
- Flask deployment
- Offline .exe packagin

User Experience (UX)
- Color-coded results
- Multilingual UI
- Drag-and-drop interface
- Visual theme/ styling
- Help guide and tips

Prioritize

2.3 Empathy Map Canvas

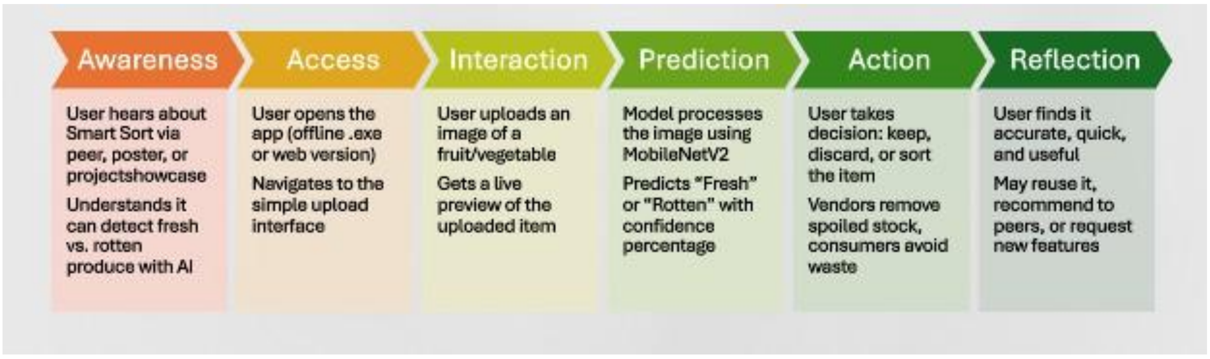
Our Empathy Map Canvas Mural:
https://app.mural.co/t/smartsort1482/m/smartsort1482/1751099737429/e037f8026d24bfc418d56ad4495985c3b05735ca?wid=193-1664821606911


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
Instantly detect fresh vs. rotten produce
Local fruit & vegetable vendors
Quality control staff at grocery chain
Farmers
Household consumers storing perishables
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Quickly identify spoiled items before display/sale
Reduce reliance on guesswork
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
"This looks fine, but it spoiled overnight"
"We don't have time to check every item"
"Why is there so much wastage?"
"Customers are rejecting our stock"
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
Frustrated with visual inspection errors
Anxious about customer trust and complaints
Worried about losing money from spoilage
GAINS
What are their wants, needs, hopes, and dreams?
Hope to increase shelf life visibility
Dream of reducing waste and boosting reputation
Want a faster, more reliable checking process
What other thoughts and feelings might influence their behavior?
Peer Influence
Cultural Beliefs
Spoiled items mixed into displays
Loss of manual sorting happening in markets
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
"I just check it by feel or smell"
"Sometimes it looks fresh but isn't"
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Often sort by guesswork or habit
Inspect produce manually before sale
Check freshness only when customers complain

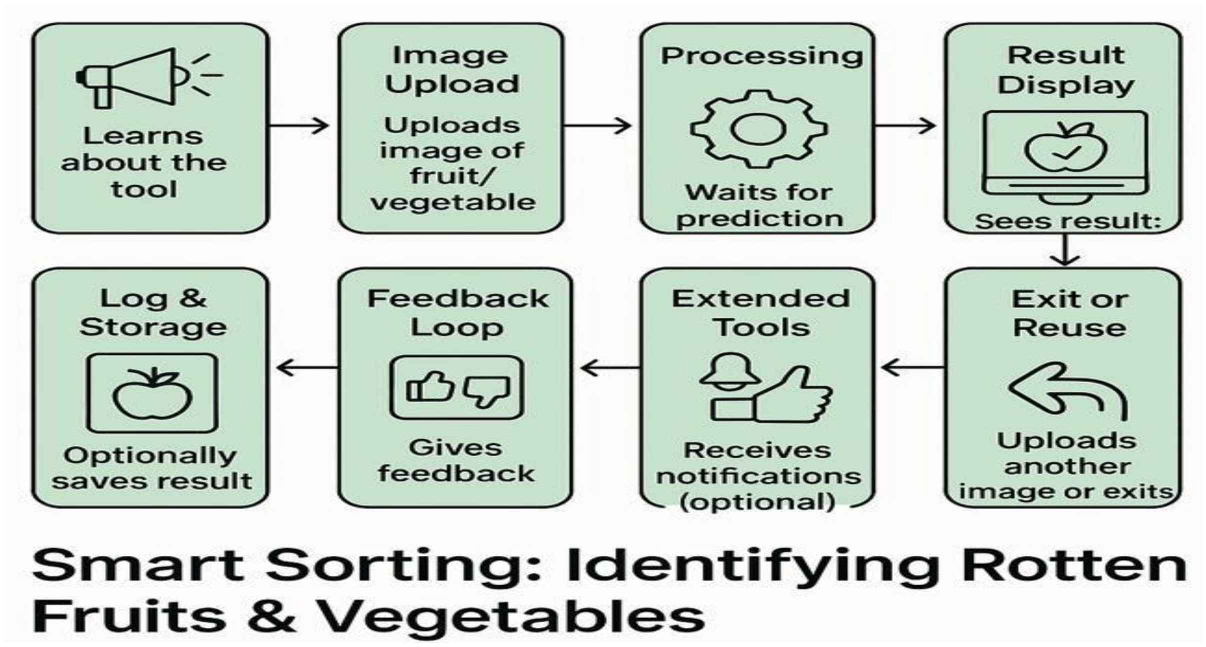
3. <u>REQUIREMENT ANALYSIS</u>

3.1 Customer Journey map

- The customer journey for **Smart Sorting** represents the typical path followed by users—from initial discovery to long-term engagement—with the system. This map helps identify key touchpoints, user needs, and emotional responses during each stage of interaction.



Awareness	Access	Interaction	Prediction	Action	Reflection
User hears about Smart Sort via peer, poster, or projectshowcase	User opens the app (offline .exe or web version)	User uploads an image of a fruit/vegetable	Model processes the image using MobileNetV2	User takes decision: keep, discard, or sort the item	User finds it accurate, quick, and useful
Understands it can detect fresh vs. rotten produce with AI	Navigates to the simple upload interface	Gets a live preview of the uploaded item	Predicts "Fresh" or "Rotten" with confidence percentage	Vendors remove spoiled stock, consumers avoid waste	May reuse it, recommend to peers, or request new features



3.2 Solution Requirement

Functional Requirements:
Following are the functional requirements of the proposed solution.

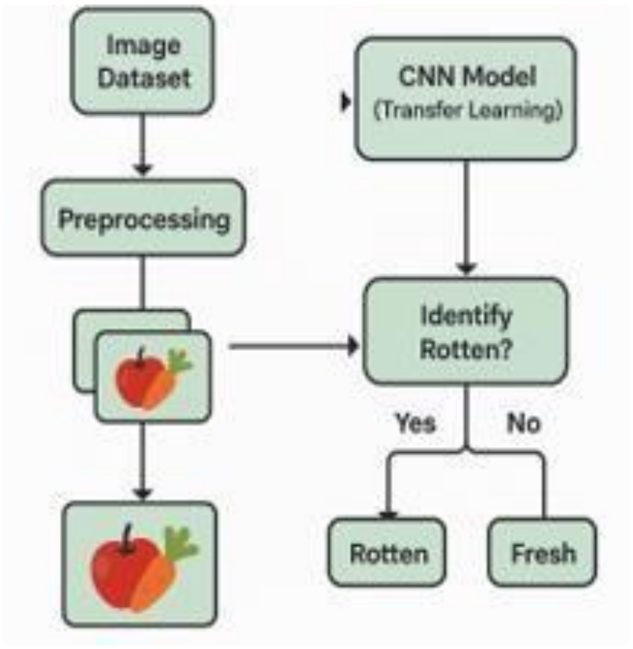
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image Upload & Prediction	Upload image through UI Get 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

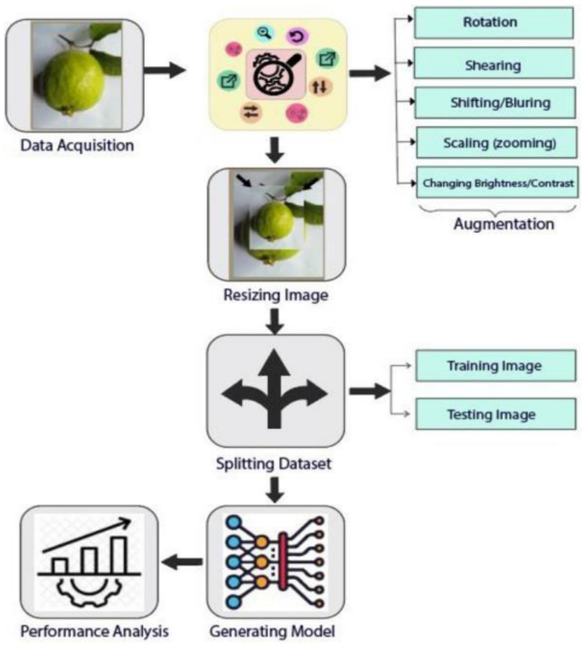
Non-functional Requirements:
Following are the non-functional requirements of the proposed solution.

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
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.	Machine Learning Model	Machine Learning model to classify fruits as fresh or rotten	MobileNetV2, Object Recognition Model, etc.
6.	Data Collection	Datasets for training the model	Kaggle

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	TensorFlow, Flask, OpenCV, Pandas	Python, Flask, TensorFlow
2.	Security Implementations	Ensures safe image handling and protects against unauthorized access or code manipulation.	Flask
3.	Scalable Architecture	Modular setup supports easy integration of future features like batch processing, Grad-CAM, or multilingual UI.	Layered architecture (Frontend–Backend–Model), reusable Python modules, transfer learning-based design
4.	Availability	Designed to operate smoothly in offline or low-connectivity environments.	PyInstaller-generated executable, local Flask server, no internet dependency.
5.	Performance	Lightweight model for fast inference, caching predictions (if needed)	MobileNetV2

4. PROJECT DESIGN

4.1 Problem Solution Fit

The **Smart Sorting** system was conceived in response to a well-defined and recurring challenge in the agricultural and food retail sectors: the difficulty of efficiently and accurately identifying rotten produce. Manual inspection methods are subjective, time-consuming, and often unreliable—especially in high-volume environments such as warehouses, markets, and supermarkets.

Our team validated the relevance of this problem by studying real-world use cases, stakeholder pain points, and gaps in existing solutions. Across customer segments—including farmers, grocery vendors, warehouse managers, and consumers—it became clear that a scalable and accessible quality assessment system was in high demand. Through iterative brainstorming and feasibility analysis, we identified that **image-based classification using transfer learning** offers a powerful and feasible pathway to address this issue. The solution is grounded in realworld utility and aligns with the technical capabilities of affordable devices. **Key Indicators of Fit**

- **Clear and Measurable Pain Points**
  Vendors lose revenue due to late detection of spoilage. Consumers discard usable produce due to visual uncertainty.
- **Existing Alternatives Are Ineffective or Inaccessible**
  Manual inspections, color sensors, and traditional rule-based systems are costly, slow, or impractical at scale.
- **Demand for Automation and Usability**
  A growing number of vendors and consumers are open to intuitive AI tools that reduce workload and improve decision quality.
- **Technically and Economically Feasible**
  Transfer learning with lightweight models such as MobileNetV2 enables deployment without high compute or cloud dependency.
- **Validated Through Personas & Use Cases**
  Problem statements were derived from roles such as supermarket inventory managers and health-conscious consumers tracking household waste.

This alignment between real-world problems and our AI-driven solution confirms the problem-solution fit and justifies continued development and deployment of the Smart Sorting system.

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: The Smart Sort Application mainly consists of:

Core Components

1. **Frontend (Presentation Layer)** ○ Built with HTML, CSS, JavaScript ○ Allows users to upload or drag-and-drop images ○ Displays prediction results (label + confidence)
2. **Backend (Application Logic)** ○ Developed using Flask (Python) ○ Handles image reception and routing
   ○ Connects to the trained deep learning model for inference
3. **Model Layer (AI Engine)** ○ Uses **MobileNetV2** (transfer learning from ImageNet) ○ Preprocessing: resize (224×224), normalize, preprocess input ○ Output: 28-class softmax prediction with high accuracy

4. **Data Layer** ○ Input dataset from Kaggle (28 classes: fresh & rotten produce) ○ Images are augmented and standardized before training ○ .h5 model file packaged with the application
5. **Deployment Layer** ○ App bundled as .exe using PyInstaller for offline use ○ Runs on standalone systems without needing Python ○ Folder structure designed for portable execution

Solution Architecture Diagram:

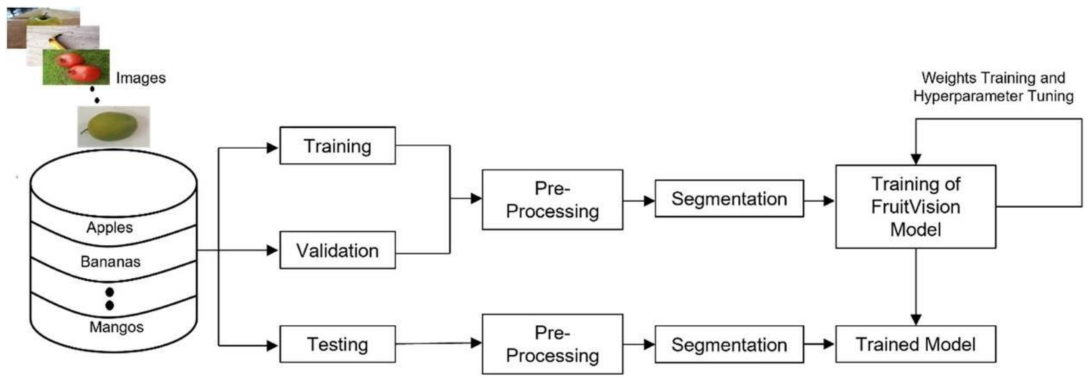


Figure 1: Architecture and data flow of the Model

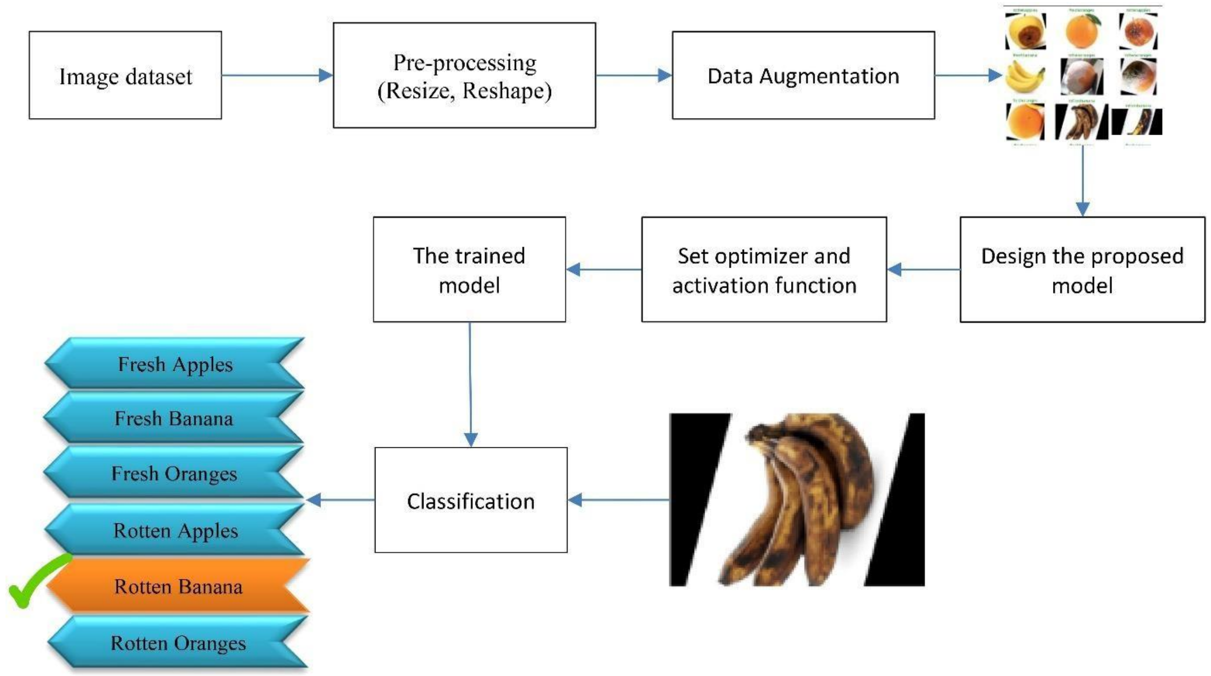


Figure 2: Architecture and data flow of the Smart Sort Application

5 PROJECT PLANNING & SCHEDULING

5.2 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	10 Feb 2026	14 Feb 2026	8	20 Feb 2026
Sprint-2	16	5 Days	15 Feb 2026	19 Feb 2026		

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:

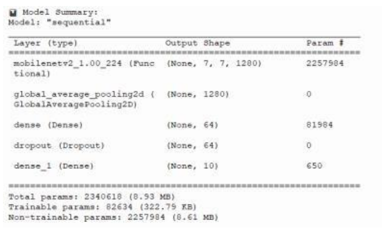
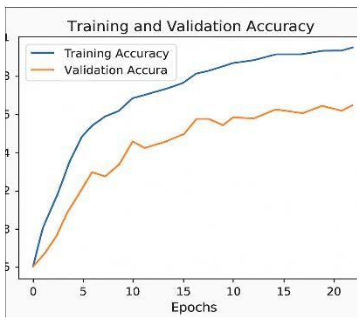
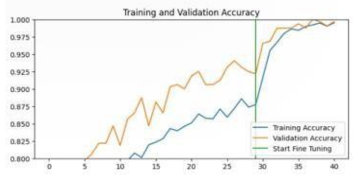
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

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

6.2 Performance Testing

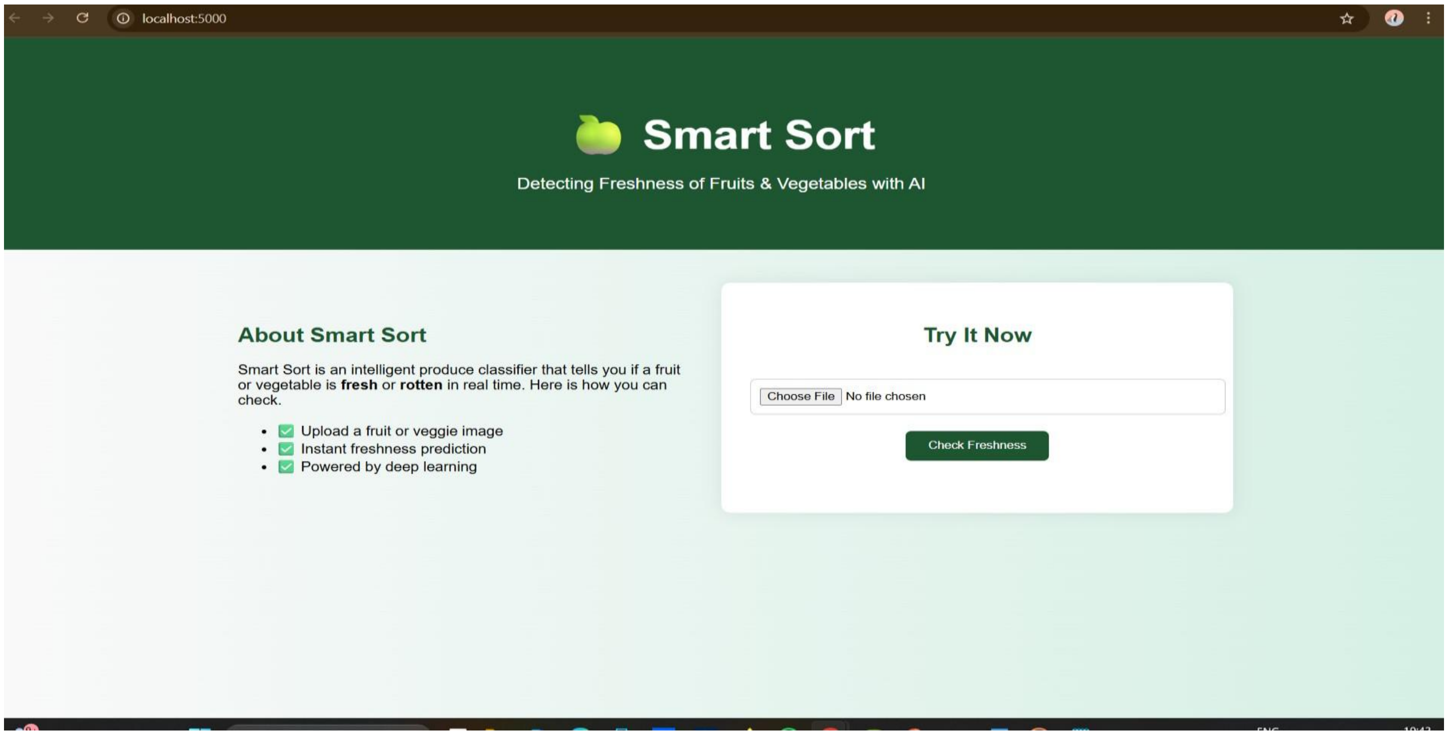
S.No.	Parameter	Values	Screenshot
	Model Summary	Transfer Learning with MobileNetV2 Input Size: 224x224 Pre-trained on ImageNet Optimizer: Adam Loss: Categorical Cross entropy	
	Accuracy	Training Accuracy: 95.6% Validation Accuracy: 94.5%	
3.	Fine Tunning Result(if Done)	Validation Accuracy: 95.8%	

```
 Model Summary:
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Func  (None, 7, 7, 1280)        2257984
 tional)

 global_average_pooling2d (  (None, 1280)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 64)                81984

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 10)                650

=================================================================
Total params: 2340618 (8.93 MB)
Trainable params: 82634 (322.79 KB)
Non-trainable params: 2257984 (8.61 MB)
```

7 RESULTS

Output Screenshots

Open the website, Click on choose file (image of fruit or vegetable) and then upload it.


localhost:5000
Smart Sort
Detecting Freshness of Fruits & Vegetables with AI

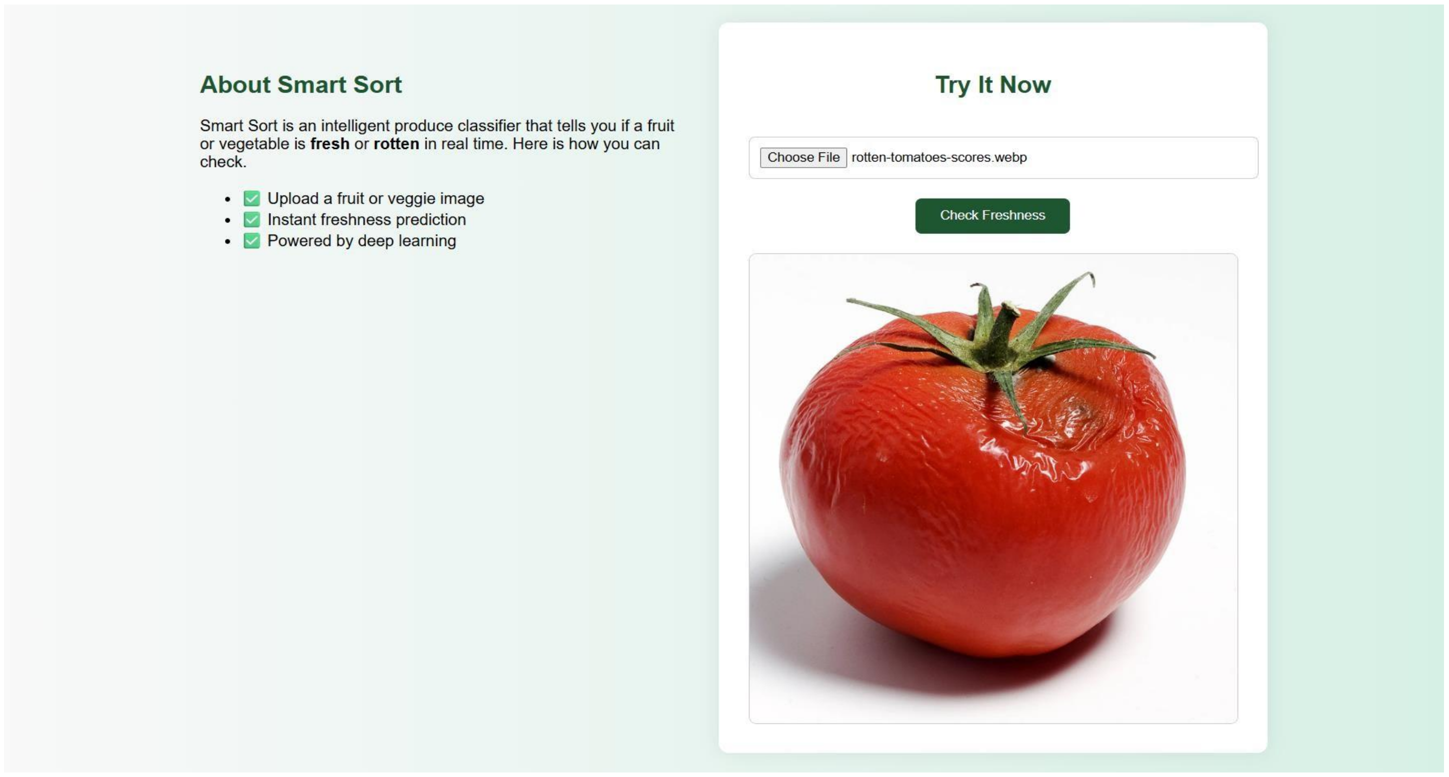
About Smart Sort

Smart Sort is an intelligent produce classifier that tells you if a fruit or vegetable is **fresh** or **rotten** in real time. Here is how you can check.

- ✓ Upload a fruit or veggie image
- ✓ Instant freshness prediction
- ✓ Powered by deep learning

Try It Now
Choose File No file chosen
Check Freshness

Now, Click on Check Freshness.

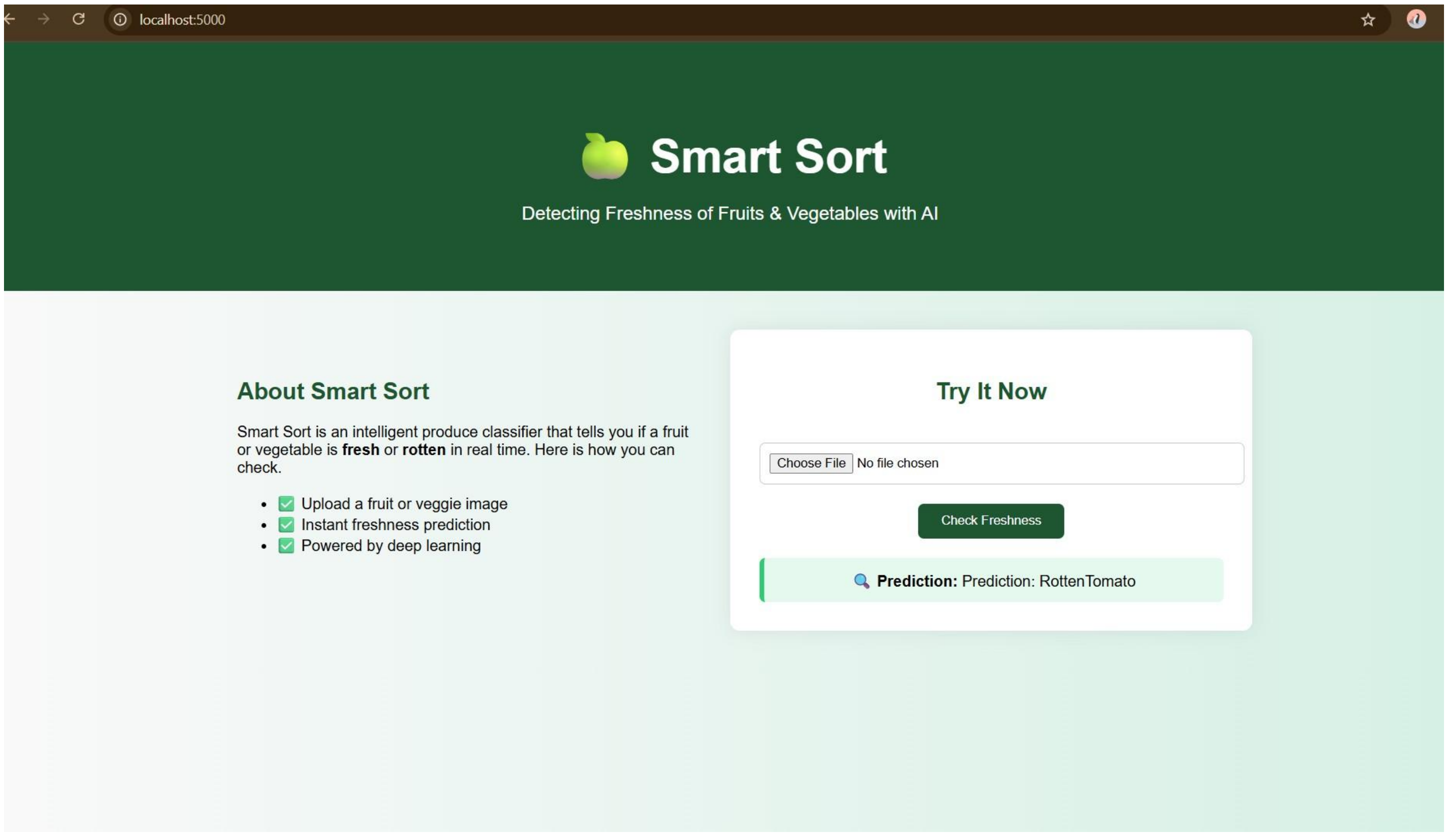


About Smart Sort

Smart Sort is an intelligent produce classifier that tells you if a fruit or vegetable is **fresh** or **rotten** in real time. Here is how you can check.

- ✓ Upload a fruit or veggie image
- ✓ Instant freshness prediction
- ✓ Powered by deep learning

The Output will be displayed as follows:


localhost:5000

Smart Sort

Detecting Freshness of Fruits & Vegetables with AI

About Smart Sort

Smart Sort is an intelligent produce classifier that tells you if a fruit or vegetable is **fresh** or **rotten** in real time. Here is how you can check.

- ✅ Upload a fruit or veggie image
- ✅ Instant freshness prediction
- ✅ Powered by deep learning

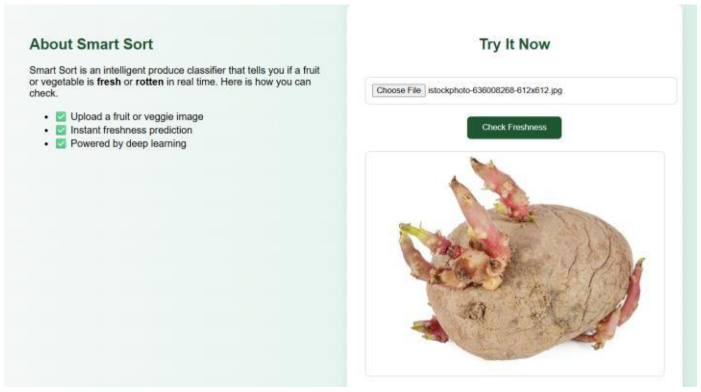
Try It Now

Choose File No file chosen

Check Freshness

🔍 **Prediction:** Prediction: RottenTomato

Similarly, For other images the respective predictions are made by Smart_sort_app.
<u>Note: The previous output will retain. So</u> <u>If you want new prediction to be made wait a while after clicking on</u> <u>Check Freshness.</u>


About Smart Sort

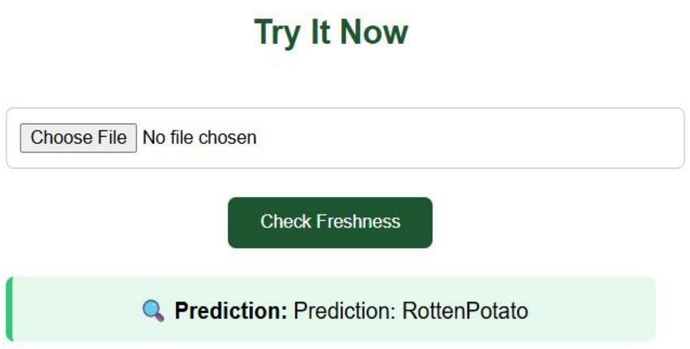
Smart Sort is an intelligent produce classifier that tells you if a fruit or vegetable is **fresh** or **rotten** in real time. Here is how you can check.

- ✅ Upload a fruit or veggie image
- ✅ Instant freshness prediction
- ✅ Powered by deep learning

Try It Now

Choose File istockphoto-636008268-612x612.jpg

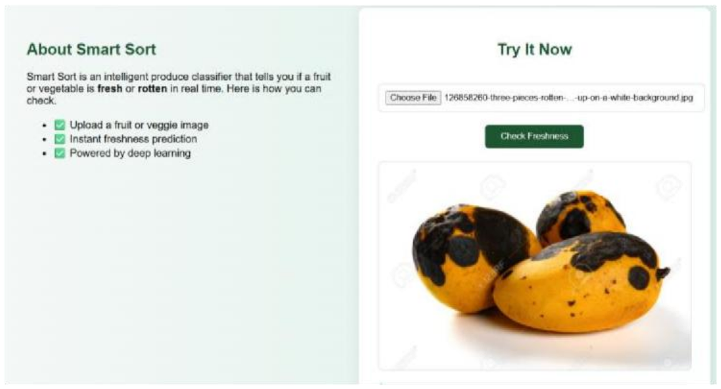
Check Freshness


Try It Now

Choose File No file chosen

Check Freshness

🔍 **Prediction:** Prediction: RottenPotato



About Smart Sort

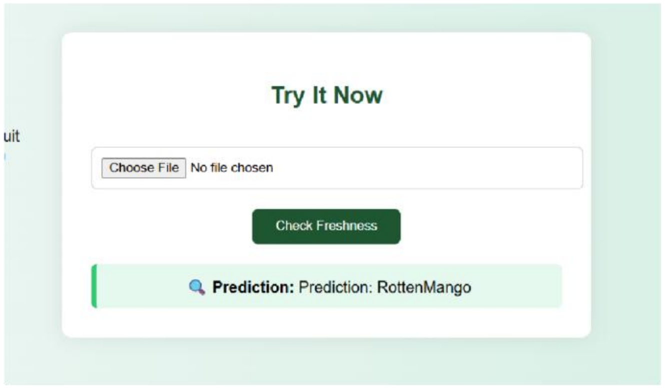
Smart Sort is an intelligent produce classifier that tells you if a fruit or vegetable is **fresh** or **rotten** in real time. Here is how you can check.

- ✓ Upload a fruit or veggie image
- ✓ Instant freshness prediction
- ✓ Powered by deep learning

Try It Now

Choose File 126858260-three-pieces-rotten-...-up-on-a-white-background.jpg

Check Freshness



Try It Now

Choose File No file chosen

Check Freshness

🔍 **Prediction:** Prediction: RottenMango

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/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten

GitHub Link:

https://github.com/KoneruNandaKishore/Smart-Sorting-Transfer-Learning-for-Identifying-Rotten-Fruits-and-Vegetables

Project Demo Link:

https://drive.google.com/file/d/1bYJT4il6PZCgebysE7HBWG-AlwE6EHwZ/view?usp=sharing