1. INTRODUCTION

1.1 Project Overview

Smart Sorting is an AI-driven project that leverages transfer learning techniques to automate the detection of rotten fruits and vegetables using image classification. The project aims to reduce the inefficiencies and inconsistencies of manual sorting by deploying a deep learning model trained to identify spoilage based on visual features.

Transfer learning allows the use of pre-trained convolutional neural networks (CNNs) (such as MobileNet, ResNet, or VGG), which are fine-tuned on a custom dataset of fresh and rotten produce. This approach minimizes the need for large labeled datasets and significantly reduces training time, while maintaining high accuracy.

The system is designed to be deployed across various platforms — from mobile applications for farmers and small vendors to automated sorting machines in food processing facilities — making it a scalable, practical, and cost-effective solution for quality control in the agricultural supply chain.

The solution has been designed with ease of deployment and scalability in mind, making it suitable for various environments such as food processing plants, distribution centers, and retail supermarkets. Through NutriGaze, businesses can significantly reduce post-harvest losses, ensure only high-quality produce reaches consumers, and enhance operational efficiency.

1.2 Purpose of the Project

The primary purpose of the NutriGaze project is to revolutionize the quality inspection process for fruits and vegetables using artificial intelligence and image-based classification. By automating the detection of rotten or spoiled produce, the system helps overcome the limitations of traditional manual sorting, thereby ensuring better food quality and reducing wastage across the supply chain.

Key Objectives of the Project:

- Eliminate manual sorting, which is time-consuming and prone to human error.
- Help meet food safety standards and regulations in supply chains.
- Help preserve fresh produce longer by removing spoiled ones early.
- To build a scalable and easy-to-use solution that can be adopted by stakeholders at various levels of the agricultural and food supply chain.
- To increase consumer confidence and satisfaction by ensuring only high-quality, fresh produce reaches the market.
- Encourage the adoption of **smart farming practices** and precision agriculture.

The NutriGaze project is not just a technological advancement but a step towards building a more sustainable, efficient, and trustworthy food supply chain.

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. PROJECT DESIGN

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

6. FUNCTIONAL AND PERFORMANCE TESTING

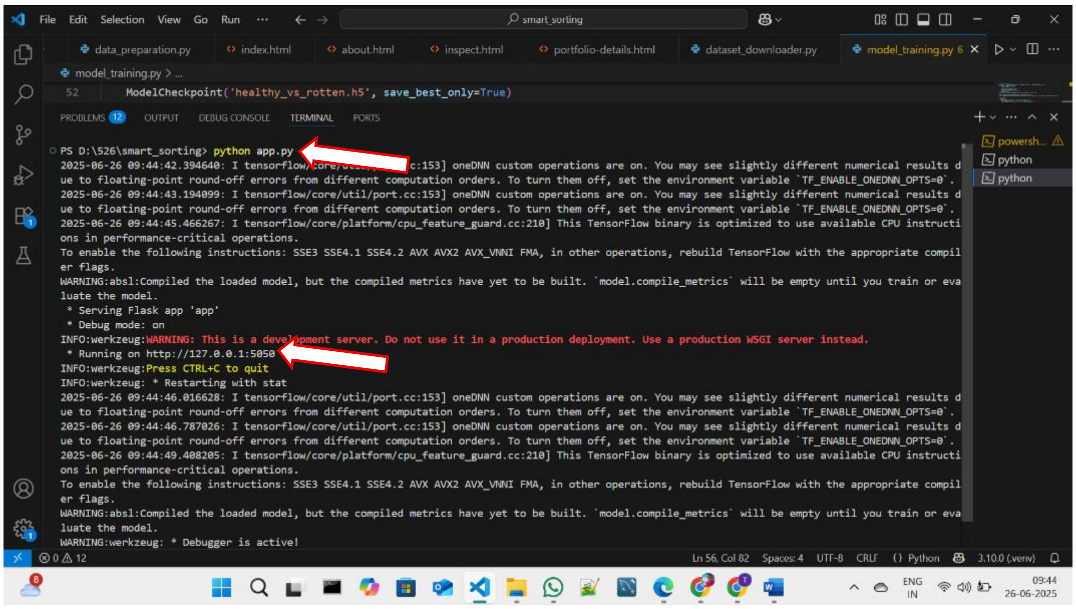
6.1 Performance Testing

7. RESULTS

7.1 Output Screenshots

The complete execution of the Smart Sorting application is shown in the images step by step as shown below.

Step 1: Run the app.py code and you will get a link in terminal as https://127.0.0.1.5050 to access web page and to do the other process.



```
ModelCheckpoint('healthy_vs_rotten.h5', save_best_only=True)
```

```
PS D:\526\smart_sorting> python app.py
2025-06-26 09:44:42.394640: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results d
ue to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-26 09:44:43.194099: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results d
ue to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-26 09:44:45.466267: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructi
ons in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compil
er flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or eva
luate the model.
 * Serving Flask app 'app'
 * Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5050
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
2025-06-26 09:44:46.016628: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results d
ue to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-26 09:44:46.787026: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results d
ue to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-26 09:44:49.408205: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructi
ons in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compil
er flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or eva
luate the model.
WARNING:werkzeug: * Debugger is active!
```

Fig 7.1.1: Code running in Terminal

Step 2: Click on that link a web page of Nutrigaze will be open in the web browser.

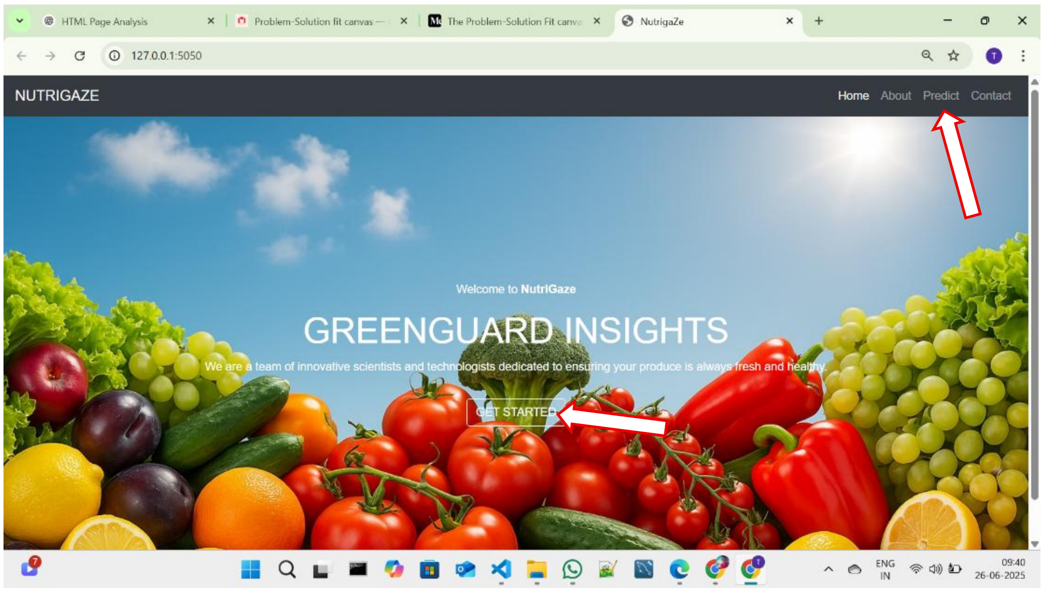

HTML Page Analysis
Problem-Solution fit canvas
The Problem-Solution Fit canva
NutrigaZe
127.0.0.1:5050
NUTRIGAZE
Home About Predict Contact
Welcome to NutriGaze
GREENGUARD INSIGHTS
We are a team of innovative scientists and technologists dedicated to ensuring your produce is always fresh and healthy.
GET STARTED
09:40
26-06-2025

Fig 7.1.2: NUTRIGAZE Home Page

Step 3: Click on GET STARTED or PREDICT option to open the prediction page.

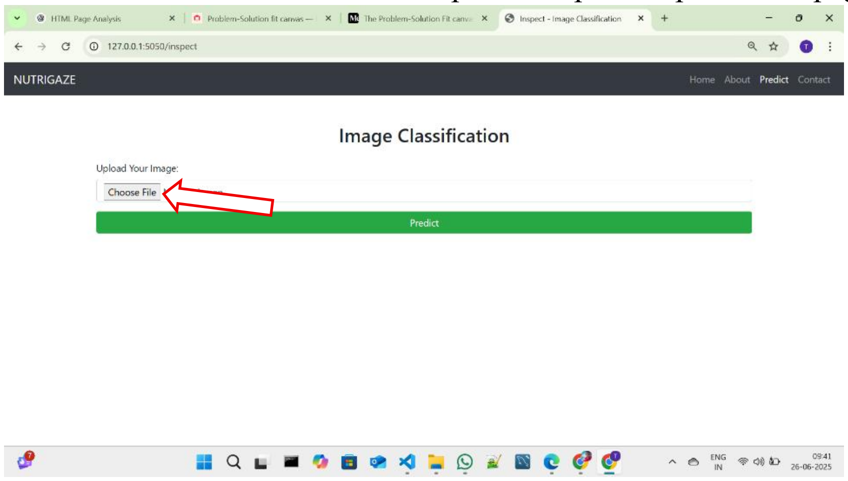


Fig 7.1.3: Prediction page in NUTRIGAGE

Step 4: Click on choose file option to choose the images that need to predict.

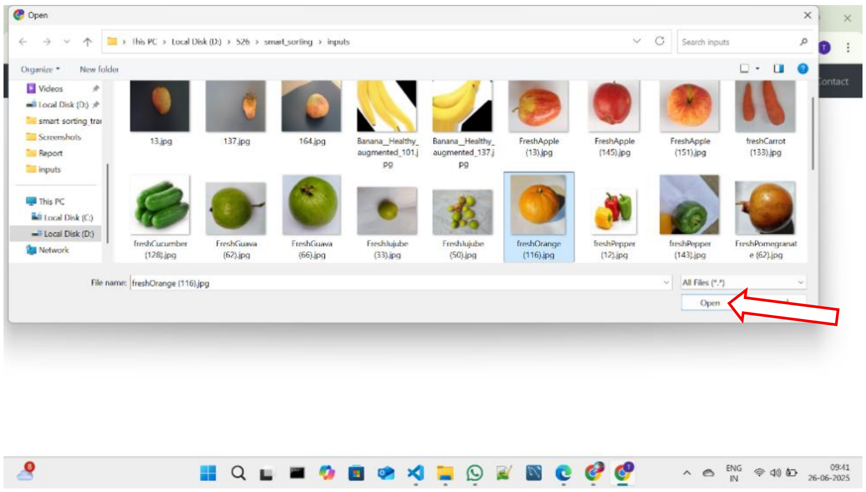


Fig 7.1.4: Window to choose image for prediction

Select any image for prediction and click on Open.

Step 5: Click on Predict to predict the quality of selected image.

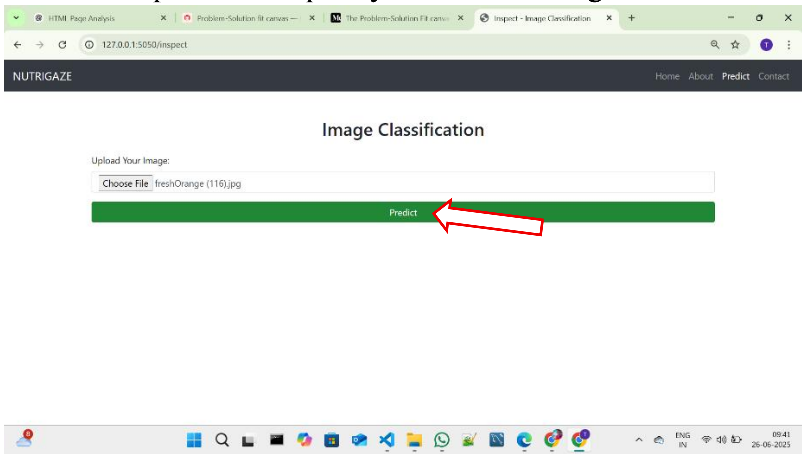


Fig 7.1.5: Image selected in prediction page

Step 6: After clicked on the predict button the model predicts the image quality and displays the quality of image.
The below images are the some of samples tested for prediction.

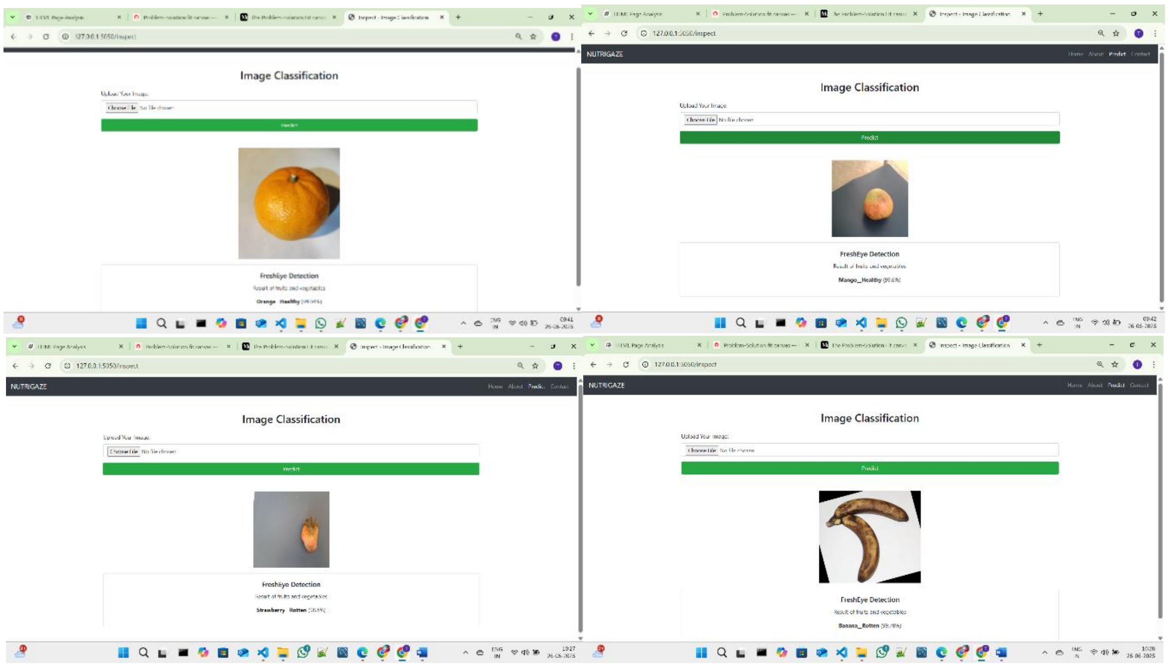


Fig 7.1.6: Prediction output for the several inputs with accuracy

After the Images Predicted the images will be stored in the uploads folder as fresh and rotten as shown in the figure given below.

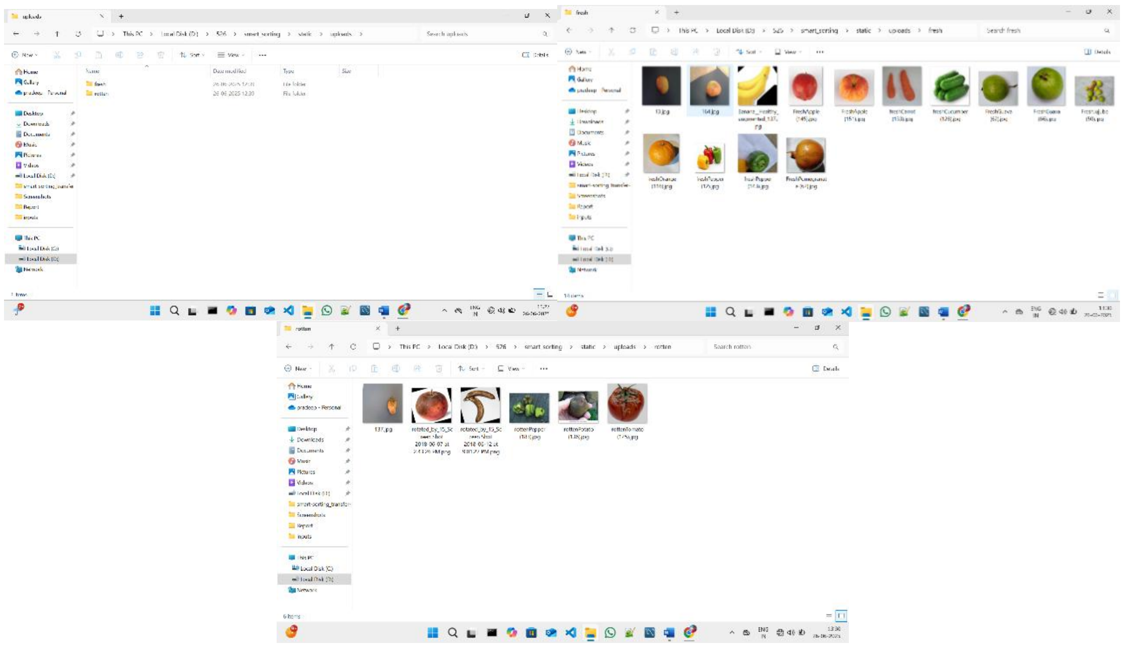

uploads
This PC > Local Disk (D:) > 526 > smart sorting > static > uploads
fresh
rotten
fresh
This PC > Local Disk (D:) > 526 > smart_sorting > static > uploads > fresh
rotten
This PC > Local Disk (D:) > 526 > smart sorting > static > uploads > rotten

Fig 7.1.7: Folder Structure to store predicted images

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages of NutriGaze Smart Sorting System

Advantage	**Description**
High Detection Accuracy	By fine-tuning pre-trained convolutional networks, you achieve more reliable spoilage detection than simple color- or rule-based methods.
Consistent, Objective Sorting	AI-driven inspection removes human subjectivity and fatigue-related errors, delivering uniform quality control across shifts and locations.
Real-Time Processing	Optimized models (e.g., MobileNet) can run inference on edge devices or smartphones, enabling instant pass/fail sorting on conveyors or in the field.
Consumer Trust & Satisfaction	Ensures only quality fruits and vegetables reach consumers, improving confidence.
Scalability & Flexibility	The same pipeline can be adapted to new fruit or vegetable types with minimal retraining, and deployed across small farms up to large processing plants.
Data-Driven Decisions	Provides objective, consistent results reducing human error.

Table 8.1: Advantages

8.2 Disadvantages / Limitations

Disadvantage	**Description**
Model Bias and Overfitting	If the training dataset is not diverse, the model may not generalize well to unseen produce or spoilage patterns.
Infrastructure Requirements	Full deployment in a commercial setting may require integration with existing machinery, cameras, and software systems, increasing setup complexity.
Limited Dataset Availability	Collecting and labeling images of various spoilage types for different fruits and vegetables can be time-consuming and costly.
External Factors	Complex external factors like hidden defects or odor cannot be detected.
Requires Technical Expertise	Maintaining, updating, and troubleshooting the AI system requires skilled personnel.

Table 8.2: Disadvantages

9. CONCLUSION

The project **"Smart Sorting"** demonstrates the practical application of **transfer learning** in addressing real-world challenges in the agricultural and food industries, particularly in the **automated detection of rotten fruits and vegetables**.

By leveraging pre-trained deep learning models and fine-tuning them with domain-specific data, the project successfully enables **accurate, fast, and consistent identification of spoiled produce**. This reduces the dependency on manual labor, minimizes food waste, enhances food safety, and improves overall supply chain efficiency.

The system proves to be **scalable and adaptable**, suitable for deployment in various settings—such as farms, warehouses, supermarkets, or mobile platforms—making it a valuable solution for modernizing quality control in agriculture.

10. FUTURE SCOPE

The Smart Sorting system can be further enhanced and expanded in several ways:

- **Support for More Categories:** Extend the model to include more fruits, vegetables, and other perishable food items.
- **Real-time Integration:** Integration with conveyor belt sorting systems for real-time industrial applications.
- **Mobile App Development:** Building a mobile version of the system for on-the-go quality checks by farmers and retailers.
- **Multi-Modal Detection:** Combining image-based detection with sensors (e.g., smell, temperature) for more comprehensive freshness checks.
- **Model Improvements:** Further fine-tuning the model with more diverse datasets to improve generalization for external images.
- **Edge Deployment:** Optimizing the model for deployment on edge devices to enable offline and on-field usage.

11. APPENDIX

Source Code:

All codes are submitted in Git-Hub Repository.

Git-Hub Repository Link:

https://github.com/NARASIMHA106525/sorting-fruits-and-vegetables

Dataset Link:

https://www.kaggle.com/datasets/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten

or
Use the dataset_downloader code in Git-Hub Repository

Project Demo Link:

https://drive.google.com/file/d/15Er4XMlVQakaXRwXQOqdOFN96thbvckt/view?usp=sharing