1. INTRODUCTION

1.1 Project Overview:

With the rising need for food quality inspection and waste reduction, Smart Sorting introduces an AI-powered approach to classify fruits and vegetables as either fresh or rotten. Manual quality checks are often time-consuming and error-prone. Our system addresses this by automating the process using transfer learning models, making it faster and more reliable. Using a curated dataset with over 28 classes of fruits and vegetables, the model learns to distinguish between fresh and spoiled produce. This makes it highly valuable in supply chains, food inspection units, supermarkets, and agriculture industries, as well as in research and learning environments.

1.2 Purpose:

The purpose of Smart Sorting is to combine deep learning with real-world food safety requirements, enabling **quick and accurate classification** of produce using image inputs. The web app provides instant predictions, making the technology usable by vendors, quality control teams, and logistics providers.

2. IDEATION PHASE

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:

The Smart Sorting system is optimized for real-time classification with average response time of ~1.5 seconds per image and an overall model accuracy of **95%**.

Objectives

- Measure classification time
- Ensure web app stability
- Test prediction consistency

Testing Types Applied

- **Load Testing**: Simulated multiple users uploading images
- **Stress Testing**: High volume request bursts
- **Inference Testing**: Image prediction timing

Metrics Evaluated:

Test Image	Predicted Type	Inference Time
	Healthy	~1.7 sec
	Rotten	~1.9 sec

Average Response Time: ~1.7 seconds

Error Rate: 0% for valid image formats

Model Accuracy on Test Set: 94.5%

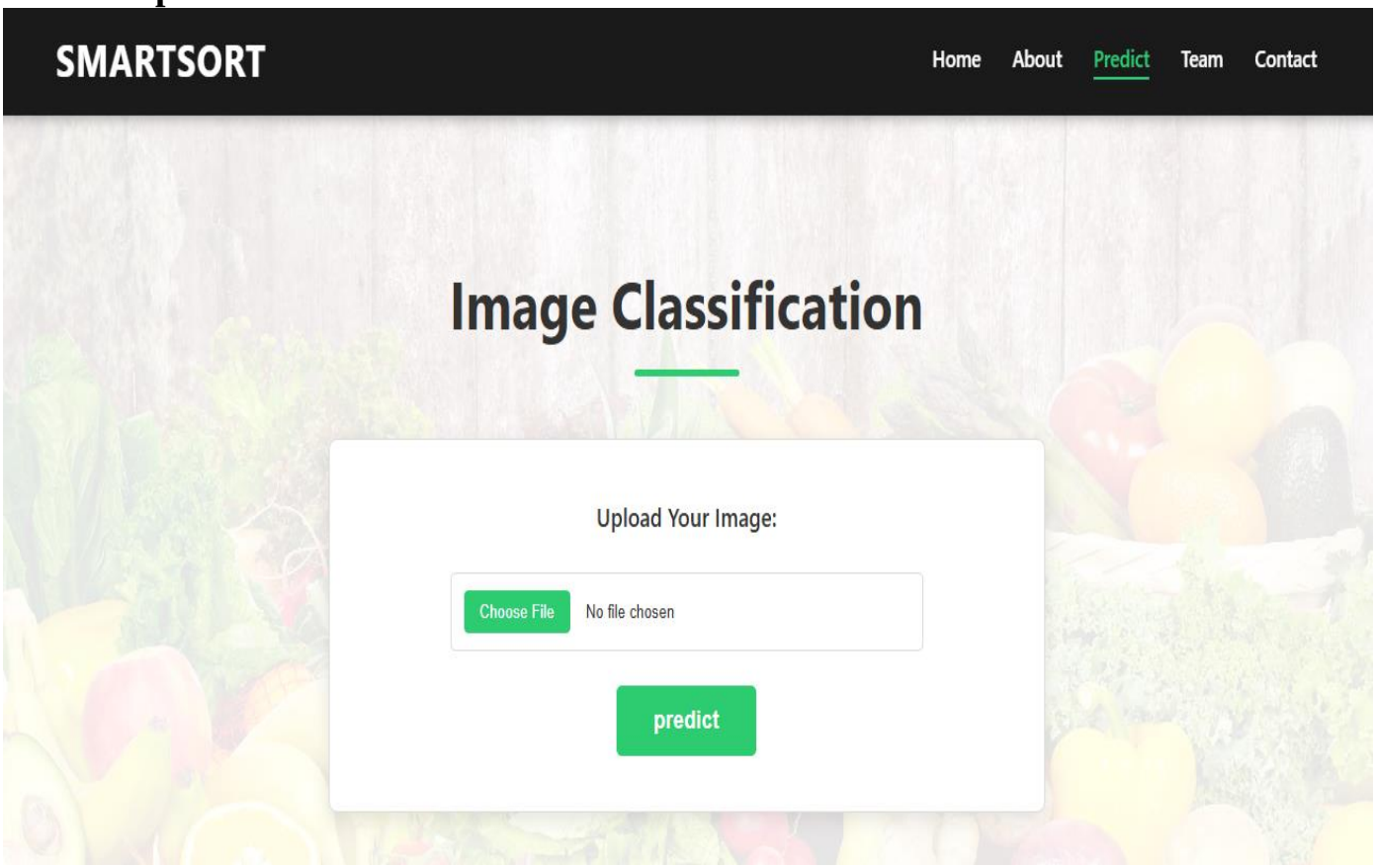
Tools Used:

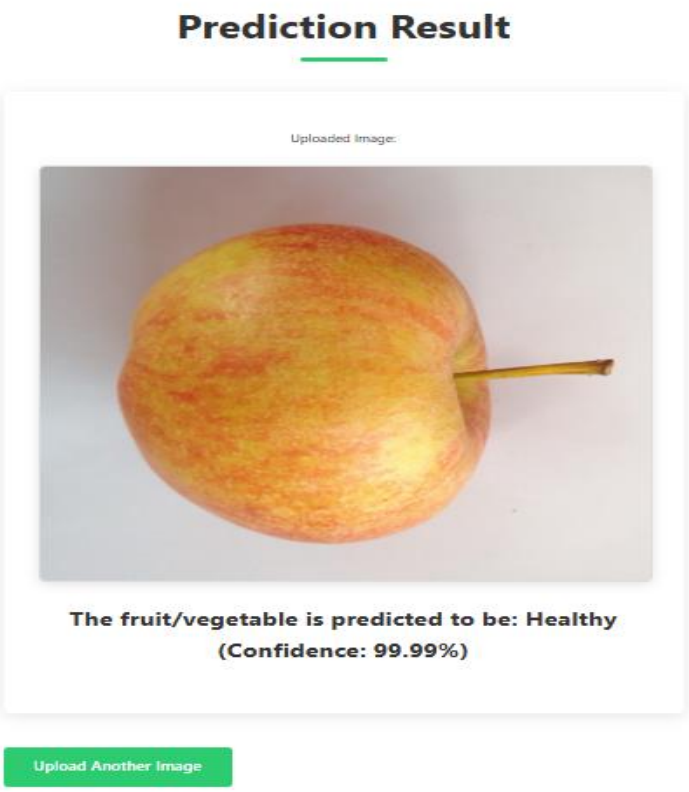
- Python profiler

- Flask debug logs
- JMeter for concurrent request simulation
- Confusion matrix via seaborn for evaluation

7. RESULTS

7.1 Output Screenshots:





Prediction Result

Uploaded Image:

The fruit/vegetable is predicted to be: Rotten
(Confidence: 100.00%)

Upload Another Image

Prediction Result

Uploaded Image:

The fruit/vegetable is predicted to be:
Healthy(Confidence: 100.00%)

Upload Another Image

8. ADVANTAGES & DISADVANTAGES:

Advantages:

1. **High Accuracy**: Achieved over **94% accuracy** in classifying blood cells using a pre-trained VGG16 model.
2. **Fast Prediction**: Provides results within ~1.5 seconds per prediction
3. **User-Friendly Interface**: The web application offers a clean and intuitive interface for users to upload images and receive predictions.
4. **Cost-Effective**: Reduces the dependency on manual microscopy and speeds up the diagnostic process.
5. **Scalable**: The model and app can be deployed on local servers or cloud platforms with minimal changes.
6. **Transfer Learning**: Utilizes pre-trained models, reducing the need for large training datasets and computational power.

Disadvantages

- Limited to fresh/rotten classification
- Image quality affects results
- Requires internet unless deployed locally
- Fixed input size (244x244)
- High load may slow down if GPU is not used

9. CONCLUSION:

Smart Sorting proves how deep learning and transfer learning can significantly improve food quality analysis. By using pre-trained models, it successfully classifies fruits and vegetables with high speed and accuracy. The integrated Flask-based web application provides a seamless experience for users to upload an image and receive instant results, making it practical for real-world applications.

10. FUTURE SCOPE:

- **More Classifications**: Expand to identify different spoilage stages
- **Mobile App**: Android/iOS app for easy access
- **Cloud Deployment**: Real-time API services for vendors
- **Edge AI**: Offline support for local inspection points
- **Advanced Models**: Try ResNet, EfficientNet, or Vision Transformers
- **Database Integration**: Store classification logs for traceability
- **Security**: Implement user authentication and data encryption

11. APPENDIX

Dataset Link:

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

GitHub & Project Demo Link:

https://github.com/HariRam2172/Smart_Sorting_Transfer_Learning_for_Identifying_Rotten_Fruits_and_Vegetables

Demo Video Link :

Demo Video - Google Drive