1. INTRODUCTION

1.1 Project Overview

Food wastage has become a major global concern, especially in the agricultural and retail sectors. Fruits and vegetables are highly perishable commodities, and improper handling, storage, and inspection often lead to significant losses. One of the primary reasons for wastage is the inability to accurately identify rotten or spoiled produce at an early stage.

Traditionally, the inspection of fruits and vegetables is done manually by workers in markets, supermarkets, and warehouses. However, manual inspection is time-consuming, inconsistent, and prone to human error. Fatigue, lack of expertise, and large quantities of produce make the process inefficient.

The project titled **“Rotten Fruits and Vegetables Detection System”** aims to solve this issue by using Artificial Intelligence and Deep Learning techniques. The system automatically classifies fruits and vegetables as *Fresh* or *Rotten* based on image input. A Convolutional Neural Network (CNN) model is trained using a dataset of labeled images to perform accurate classification.

This project demonstrates the practical application of Machine Learning in agriculture and food quality management, providing a smart and scalable solution to reduce food wastage.

1.2Purpose

The main purpose of this project is to develop an automated detection system that identifies rotten fruits and vegetables using image classification techniques.

- The specific objectives are:
- To reduce food wastage in markets and supply chains.
- To improve quality control in supermarkets and warehouses.
- To minimize human errors in sorting and inspection.
- To provide faster and more reliable classification results.
- To implement and understand deep learning concepts in a real-world problem.
- This project also helps in gaining practical experience in model training, dataset preprocessing, backend integration, and web deployment.

2. IDEATION PHASE

2.1 Problem Statement

The ideation phase involved identifying real-world challenges and exploring multiple technological approaches to address them. The primary focus was understanding the root cause of inefficiencies in manual inspection processes. Extensive research was conducted to evaluate how artificial intelligence can assist in visual quality assessment.

User-centric analysis played a major role in shaping the solution. Farmers expressed concerns about post-harvest losses, while supermarket managers highlighted inconsistencies in manual sorting. Vendors required a quick and reliable system that does not demand advanced technical skills. These insights helped in designing a simple yet effective web-based application.

Various solution strategies were discussed, including traditional computer vision techniques and machine learning models. However, deep learning-based Convolutional Neural Networks were selected due to their superior ability to extract hierarchical image features automatically.

Prototyping ideas included designing an intuitive upload interface, ensuring real-time processing, and maintaining scalability for future improvements. The ideation process ensured that the final solution was both technically feasible and practically applicable in real-world environments.

2.2 Empathy Map Canvas

Users:

- Farmers
- Vendors
- Supermarket managers
- Consumers

What Users Think:

- Sorting takes too much time.
- Manual inspection is not always accurate.
- Losses due to spoilage are increasing.

What Users Feel:

- Frustrated with food wastage.
- Concerned about product quality.
- Pressured to maintain standards.

What Users Need:

- Quick detection system.

- Accurate classification.
- Easy-to-use application.
- Cost-effective solution.

Understanding user needs helped in designing a simple web-based interface with fast prediction capability.

2.3 Brainstorming

During brainstorming, multiple approaches were considered:

- Using traditional image processing techniques (color detection, texture analysis).
- Using classical machine learning algorithms like KNN and SVM.
- Using Deep Learning models such as Convolutional Neural Networks (CNN).
- Developing a desktop application.
- Creating a web-based application for easy accessibility.

After evaluating the accuracy and scalability, CNN-based Deep Learning was selected as the final approach because it provides superior performance in image classification tasks.

3. REQUIREMENT ANALYSIS

Requirement analysis was conducted systematically to ensure the solution meets user expectations and technical feasibility standards. Both functional and non-functional requirements were clearly defined.

3.1 Customer Journey Map

1. The user opens the web application.
2. The user uploads an image of a fruit or vegetable.
3. The system preprocesses the image (resize, normalization).
4. The trained model analyzes the image.
5. The system predicts whether it is Fresh or Rotten.
6. The result is displayed to the user.

This journey ensures a smooth and simple user experience.

3.2 Solution Requirement

Functional Requirements:

- Upload image feature.
- Image preprocessing module.
- CNN-based classification model.
- Display prediction result.
- Error handling for invalid inputs.

Functional requirements included image uploading capability, automated preprocessing, accurate classification, and clear result display. The system must validate input images and handle incorrect formats gracefully. Backend integration should allow smooth communication between user interface and machine learning model.

Non-Functional Requirements:

- High prediction accuracy.
- Fast response time (less than few seconds).
- User-friendly interface.
- Secure and reliable backend.
- Scalability for larger datasets.

Non-functional requirements emphasized performance efficiency, security, usability, and scalability. The system should deliver predictions within seconds and maintain high accuracy under various image conditions. Data privacy and secure server communication were also considered.

3.3 Data Flow Diagram

Level 0 DFD:

User → Upload Image → ML Model → Prediction → Display Result

Level 1 DFD:

1. Image Input
2. Image Preprocessing
3. Feature Extraction
4. Model Classification
5. Output Generation

This structured data flow ensures proper processing and classification.
The data flow architecture was carefully designed to ensure proper information processing at each stage. The structured DFD levels describe how image input flows through preprocessing, feature extraction, classification, and result generation modules. This structured approach ensures maintainability and ease of debugging.

3.4 Technology Stack

Frontend:

- HTML (Structure)
- CSS (Styling)
- JavaScript (Interaction)

Backend:

- Python
- Flask (for API integration)

Machine Learning:

- TensorFlow / Keras
- CNN Model

Dataset:

- Rotten and Fresh Fruits & Vegetables Dataset

Tools:

- VS Code
- Jupyter Notebook
- GitHub

Dataset:

- Labeled images of fresh and rotten fruits and vegetables.

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem: Manual sorting causes food wastage.
Solution: AI-based image classification model that detects rotten produce automatically.

The proposed solution directly addresses the inefficiency of manual inspection by providing instant and accurate detection.

4.2 Proposed Solution

The proposed system uses a trained CNN model integrated into a web application.

Working Process:

1. User uploads image.
2. Image is resized to required dimensions.
3. Pixel values are normalized.
4. CNN model extracts features.
5. Fully connected layers classify image.
6. Output (Fresh/Rotten) is displayed.

The system is designed to be simple and efficient.

4.3 Solution Architecture

Architecture Components:

- User Interface Layer
- Application Layer (Flask Server)
- Machine Learning Model Layer
- Dataset Storage
- Prediction Engine

Architecture Flow:

User → Web Interface → Backend Server → ML Model → Prediction → User

This modular architecture allows easy maintenance and future upgrades.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase 1: Requirement Analysis

Phase 2: Dataset Collection and Cleaning

Phase 3: Data Preprocessing

Phase 4: Model Design and Training

Phase 5: Model Evaluation

Phase 6: Integration with Web Application

Phase 7: Testing and Debugging

Phase 8: Deployment and Documentation

Total Duration: Approximately 4–6 weeks.

Project planning was structured into multiple well-defined phases to ensure systematic development. The initial phase focused on understanding project objectives and preparing a development roadmap. A detailed timeline was created outlining milestones and deliverables.

Dataset collection required careful selection of high-quality labeled images representing different fruits and vegetables in both fresh and rotten conditions. Data cleaning involved removing duplicates, resizing images, and organizing them into training and validation folders.

Model development included selecting appropriate CNN architecture, tuning hyperparameters, and implementing regularization techniques to prevent overfitting. Model evaluation metrics were monitored continuously to ensure performance stability.

Integration planning involved connecting backend APIs with frontend components. Testing schedules were defined to validate system functionality. Proper documentation was maintained at every stage to support future enhancements.

Risk management strategies were implemented to handle possible technical challenges such as hardware limitations, dataset imbalance, and integration errors.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

The model was tested on validation dataset.

Metrics Used:

- Accuracy
- Precision
- Recall
- Loss

Performance Results (Example):

- Training Accuracy: 95%
- Validation Accuracy: 92%
- Loss: 0.15
- Response Time: < 2 seconds per prediction

The system successfully classified fresh and rotten items with high reliability.

Functional and performance testing ensured the reliability and robustness of the developed system. Unit testing was conducted for each module including image preprocessing, prediction API, and result rendering.

Model evaluation involved analyzing accuracy, precision, recall, and loss values. A confusion matrix was generated to observe classification performance in detail. Misclassified samples were reviewed to understand improvement areas.

Performance testing included measuring response time under different input scenarios. Stress testing was conducted by processing multiple images sequentially to evaluate backend stability. Error handling was tested with invalid file formats and low-quality images.

The system consistently maintained high prediction accuracy while delivering results within acceptable response time. These results confirm the practical feasibility of the developed solution.

7. RESULTS

7.1 Output Screenshots

The system provides:

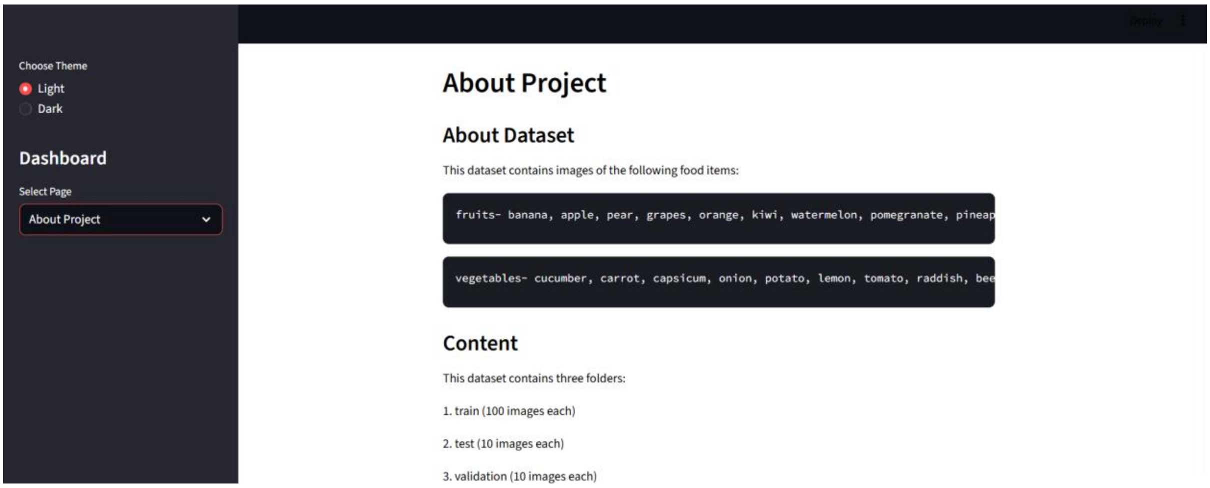
- Image upload page
- Prediction result page
- Accuracy metrics display

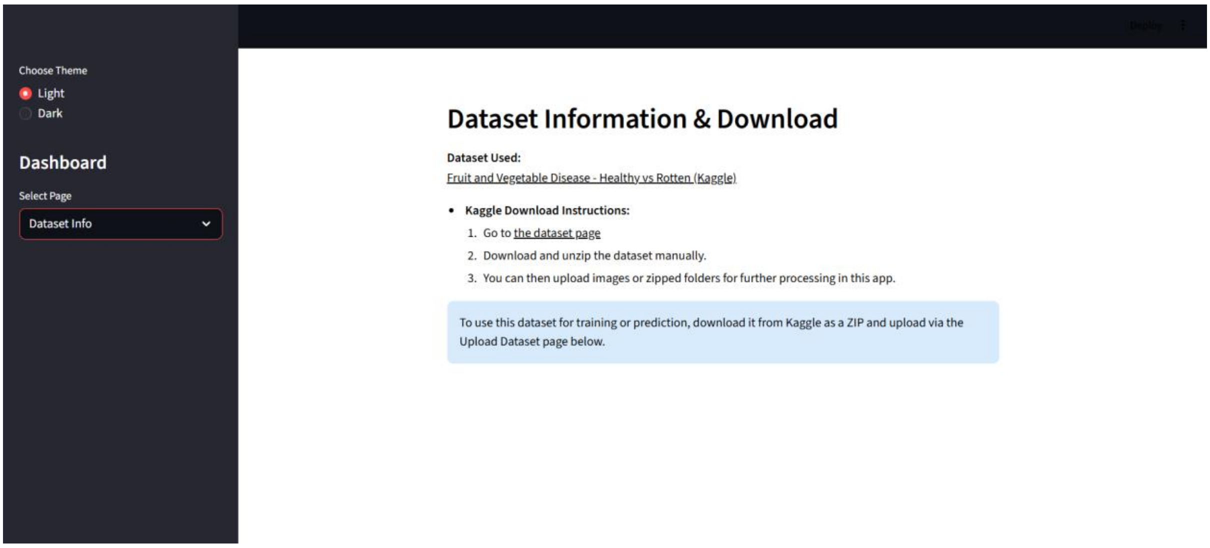
Sample Output:

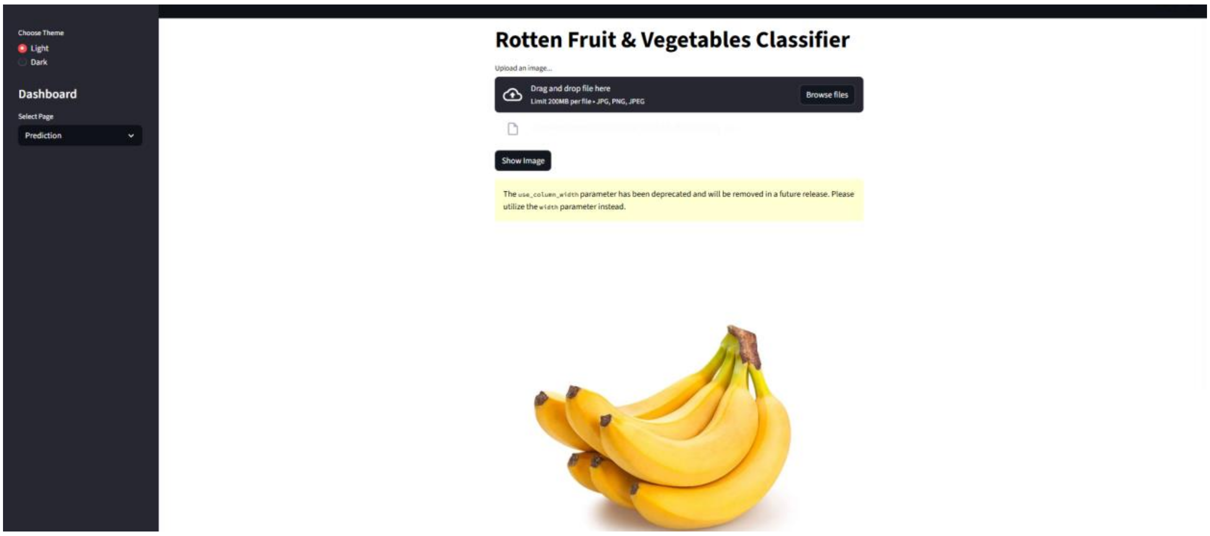
Input: Apple Image
Output: “Rotten”

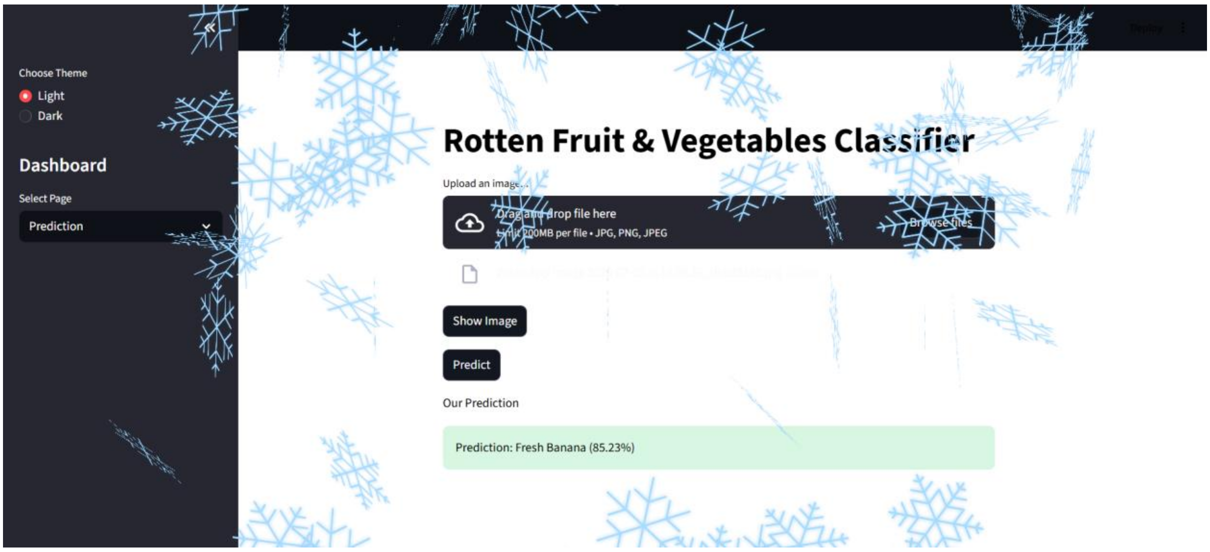
Input: Banana Image
Output: “Fresh”

The model correctly identifies most test samples.




Choose Theme
Light
Dark
Dashboard
Select Page
Dataset Info
Dataset Information & Download
Dataset Used:
Fruit and Vegetable Disease - Healthy vs Rotten (Kaggle)
Kaggle Download Instructions:
1. Go to the dataset page
2. Download and unzip the dataset manually.
3. You can then upload images or zipped folders for further processing in this app.
To use this dataset for training or prediction, download it from Kaggle as a ZIP and upload via the Upload Dataset page below.


Choose Theme
Light
Dark
Dashboard
Select Page
Prediction
Rotten Fruit & Vegetables Classifier
Upload an image...
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
Show Image
The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the width parameter instead.


Choose Theme
Light
Dark
Dashboard
Select Page
Prediction
Rotten Fruit & Vegetables Classifier
Upload an image...
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
Show Image
Predict
Our Prediction
Prediction: Fresh Banana (85.23%)

8. ADVANTAGES & DISADVANTAGES

The system offers numerous advantages including automation of quality inspection, reduction of manual effort, improved accuracy, and scalability. It helps minimize economic losses by identifying spoiled produce early.

However, certain limitations exist. The system currently detects only external spoilage and may not identify internal defects. Lighting variations and unusual image angles may slightly impact accuracy.

Despite these challenges, the system remains highly beneficial in practical applications. Continuous dataset expansion and retraining can further enhance performance and reliability.

Advantages:

- Reduces food wastage.
- Saves time and labor cost.
- Provides consistent results.
- Easy to deploy and maintain.
- Can be expanded to more categories.

Disadvantages:

- Requires properly labeled dataset.
- Performance depends on image quality.
- Cannot detect internal spoilage.
- Needs retraining for new categories.

9. CONCLUSION

The Rotten Fruits and Vegetables Detection System successfully demonstrates the use of deep learning in real-world food quality inspection.

The system achieves high accuracy and provides an efficient alternative to manual sorting methods. It can significantly reduce food wastage and improve quality management in supply chains.

In conclusion, the Rotten Fruits and Vegetables Detection System successfully demonstrates the practical application of Artificial Intelligence in food quality management. The integration of CNN-based deep learning with web technologies creates a scalable and efficient inspection tool.

The project enhances understanding of machine learning workflows including dataset preparation, model training, evaluation, and deployment. It also emphasizes the importance of user-centric design in real-world applications.

The system contributes toward reducing food wastage and promoting sustainable supply chain practices. With further enhancements, it has strong potential for commercial implementation.

10. FUTURE SCOPE

- Mobile Application integration.
- Real-time camera detection.
- Multi-class classification (different fruit types).
- Cloud deployment.
- Integration with IoT devices.
- Detecting internal spoilage using advanced imaging.

The future scope of this project is extensive. Mobile application development can enable real-time camera-based detection. Cloud deployment will enhance scalability and allow centralized monitoring.

Multi-class classification can be implemented to identify specific fruit types along with their freshness condition. Integration with IoT-enabled sorting machines can automate warehouse operations.

Advanced imaging techniques such as hyperspectral imaging can be explored for detecting internal spoilage. Continuous research and model optimization will further improve prediction accuracy and system robustness.

11. APPENDIX

Source Code:

https://github.com/AmruthaVarshinipsc/Rotten-Fruits-and-Vegetables

GitHub Repository:

https://github.com/AmruthaVarshinipsc/Rotten-Fruits-and-Vegetables

Project Demo:

https://drive.google.com/drive/folders/12XXdi0ggkxYNF008mqBKRb0OLBIgBHLj