Project Report Format :

1. Introduction:

1.1 Project Overview:

Title:

Smart Sorting: Transfer learning for identifying rotten fruits and vegetables

Objective:

The main objective of the **Fruit Freshness Detection System** project is to develop an automated image-based classification system that accurately identifies whether a fruit is fresh or rotten using deep learning techniques.
The system aims to reduce manual inspection effort, improve accuracy in quality assessment, and provide a fast, reliable, and user-friendly solution that can be deployed as a web application for real-time usage.

1.2 Purpose

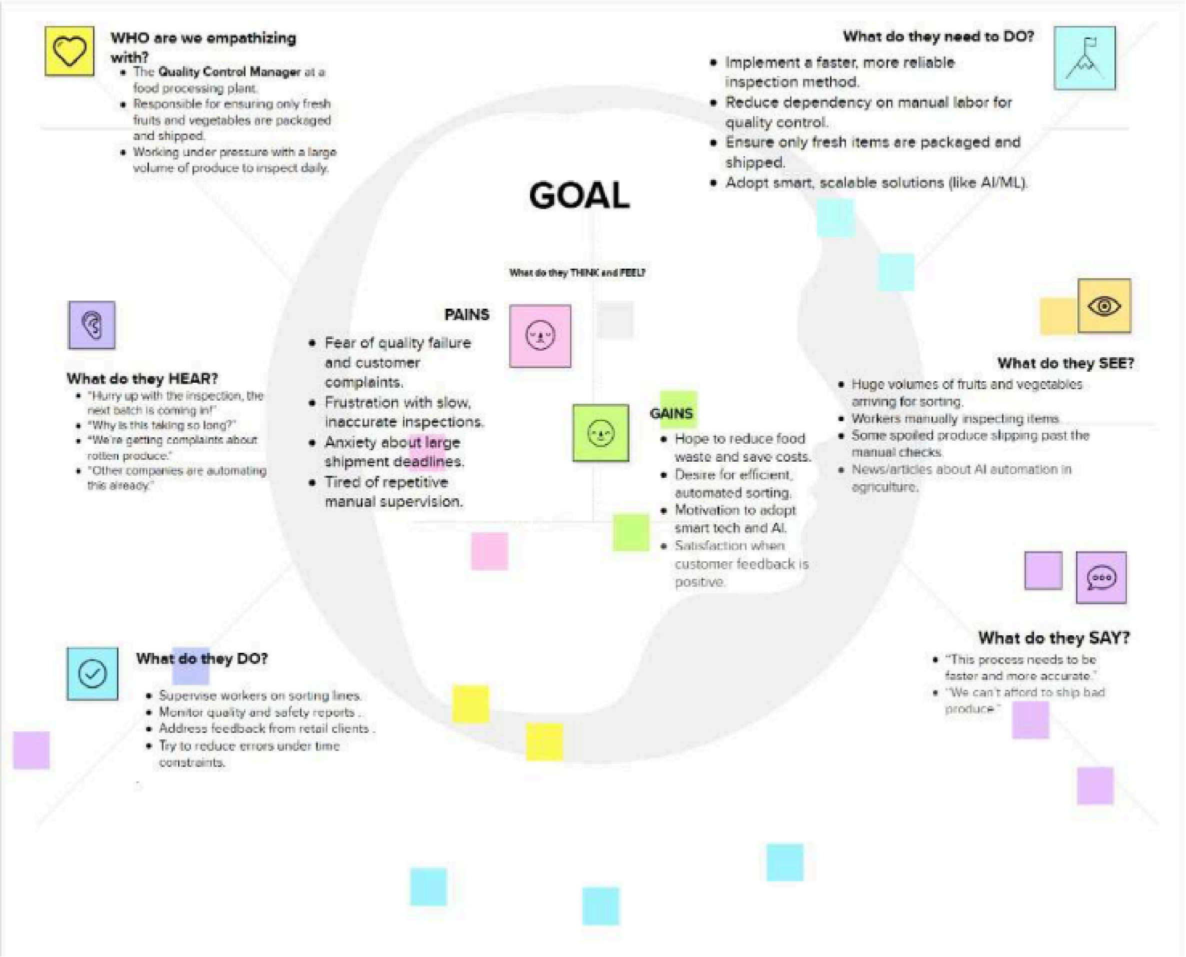
The purpose of this project is to build an intelligent system that can automatically detect whether a fruit is fresh or rotten using image processing and deep learning techniques.
It is designed to improve quality control, reduce human error, save time, and support automation in industries such as agriculture, supermarkets, and food supply management.

2. Ideation Phase

2.1 Problem Statement

Manual inspection of fruits to determine freshness is time-consuming, inconsistent, and prone to human error.There is a need for an automated, accurate, and reliable system that can classify fruits as fresh or rotten using image-based deep learning techniques.

2.2 Empathy Map Canvas



WHO are we empathizing with?

- The **Quality Control Manager** at a food processing plant.
- Responsible for ensuring only fresh fruits and vegetables are packaged and shipped.
- Working under pressure with a large volume of produce to inspect daily.

What do they need to DO?

- Implement a faster, more reliable inspection method.
- Reduce dependency on manual labor for quality control.
- Ensure only fresh items are packaged and shipped.
- Adopt smart, scalable solutions (like AI/ML).

GOAL

What do they THINK and FEEL?

PAINS

- Fear of quality failure and customer complaints.
- Frustration with slow, inaccurate inspections.
- Anxiety about large shipment deadlines.
- Tired of repetitive manual supervision.

GAINS

- Hope to reduce food waste and save costs.
- Desire for efficient, automated sorting.
- Motivation to adopt smart tech and AI.
- Satisfaction when customer feedback is positive.

What do they HEAR?

- "Hurry up with the inspection, the next batch is coming in!"
- "Why is this taking so long?"
- "We're getting complaints about rotten produce."
- "Other companies are automating this already."

What do they SEE?

- Huge volumes of fruits and vegetables arriving for sorting.
- Workers manually inspecting items.
- Some spoiled produce slipping past the manual checks.
- News/articles about AI automation in agriculture.

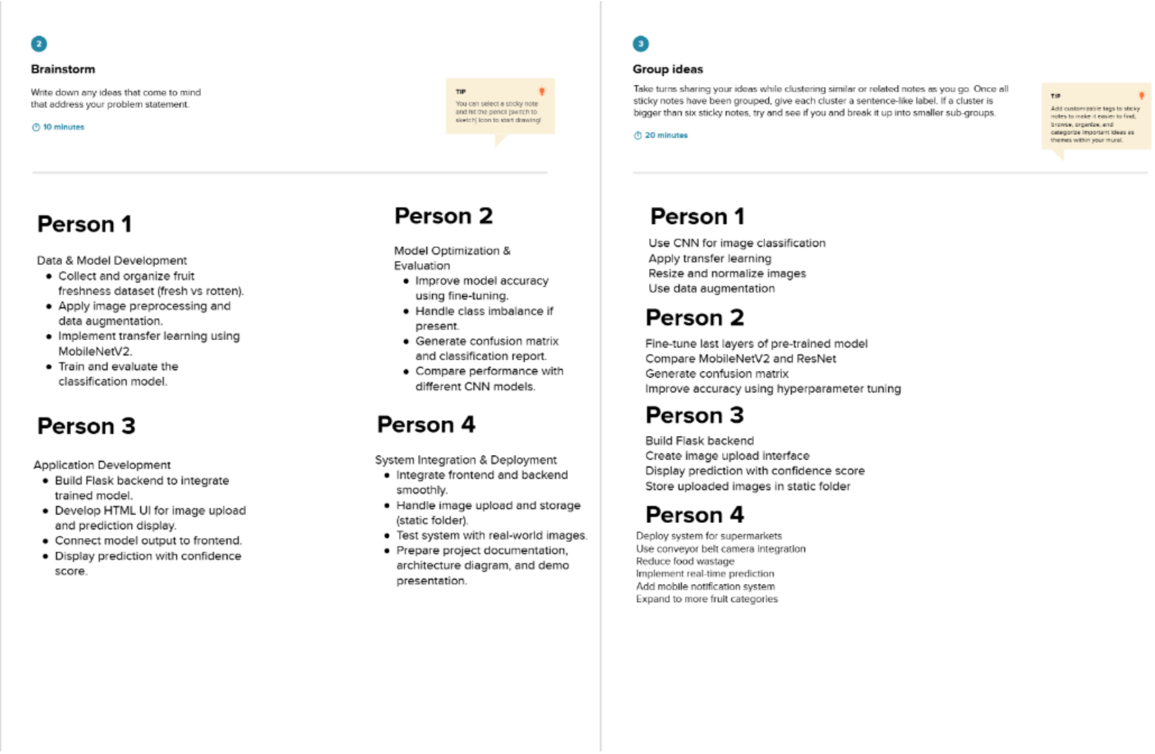
What do they DO?

- Supervise workers on sorting lines.
- Monitor quality and safety reports .
- Address feedback from retail clients .
- Try to reduce errors under time constraints.

What do they SAY?

- "This process needs to be faster and more accurate."
- "We can't afford to ship bad produce."

2.3 Brainstorming



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Data & Model Development

- Collect and organize fruit freshness dataset (fresh vs rotten).
- Apply image preprocessing and data augmentation.
- Implement transfer learning using MobileNetV2.
- Train and evaluate the classification model.

Person 2

Model Optimization & Evaluation

- Improve model accuracy using fine-tuning.
- Handle class imbalance if present.
- Generate confusion matrix and classification report.
- Compare performance with different CNN models.

Person 3

Application Development

- Build Flask backend to integrate trained model.
- Develop HTML UI for image upload and prediction display.
- Connect model output to frontend.
- Display prediction with confidence score.

Person 4

System Integration & Deployment

- Integrate frontend and backend smoothly.
- Handle image upload and storage (static folder).
- Test system with real-world images.
- Prepare project documentation, architecture diagram, and demo presentation.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Person 1

Use CNN for image classification
Apply transfer learning
Resize and normalize images
Use data augmentation

Person 2

Fine-tune last layers of pre-trained model
Compare MobileNetV2 and ResNet
Generate confusion matrix
Improve accuracy using hyperparameter tuning

Person 3

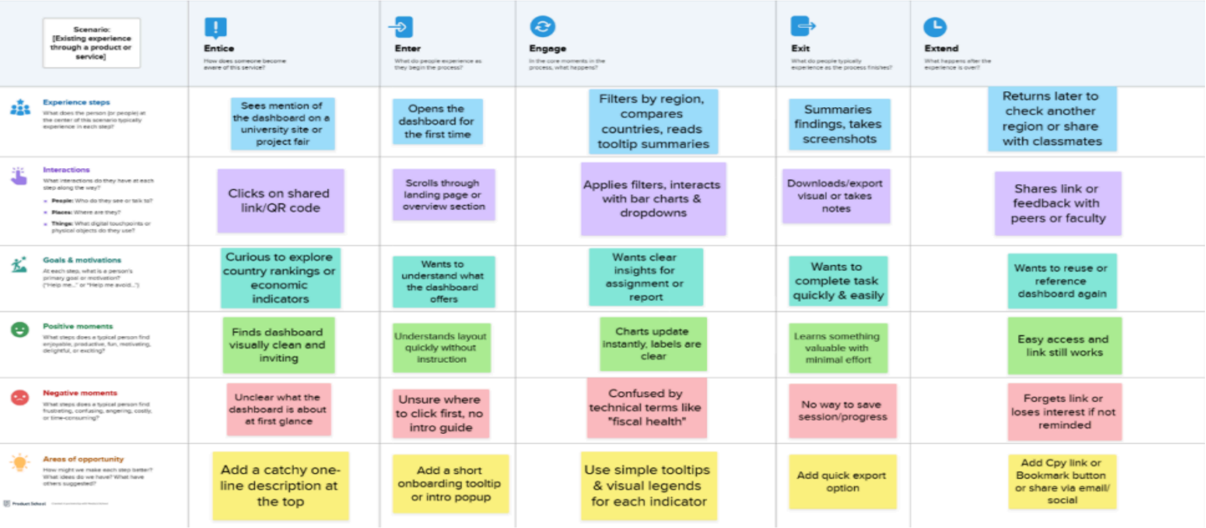
Build Flask backend
Create image upload interface
Display prediction with confidence score
Store uploaded images in static folder

Person 4

Deploy system for supermarkets
Use conveyor belt camera integration
Reduce food wastage
Implement real-time prediction
Add mobile notification system
Expand to more fruit categories

3. Requirement Analysis

3.1 Customer Journey Map



Scenario: [Existing experience through a product or service]	**Entice** How does someone become aware of this service?	**Enter** What do people experience as they begin the process?	**Engage** In the core moments in the process, what happens?	**Exit** What do people typically experience as the process finishes?	**Extend** What happens after the experience is over?
Experience steps What does the person (or people) at the center of this scenario typically experience in each step?	Sees mention of the dashboard on a university site or project fair	Opens the dashboard for the first time	Filters by region, compares countries, reads tooltip summaries	Summaries findings, takes screenshots	Returns later to check another region or share with classmates
Interactions What interactions do they have at each step along the way? • **People:** Who do they see or talk to? • **Places:** Where are they? • **Things:** What digital touchpoints or physical objects do they use?	Clicks on shared link/QR code	Scrolls through landing page or overview section	Applies filters, interacts with bar charts & dropdowns	Downloads/export visual or takes notes	Shares link or feedback with peers or faculty
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	Curious to explore country rankings or economic indicators	Wants to understand what the dashboard offers	Wants clear insights for assignment or report	Wants to complete task quickly & easily	Wants to reuse or reference dashboard again
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Finds dashboard visually clean and inviting	Understands layout quickly without instruction	Charts update instantly, labels are clear	Learns something valuable with minimal effort	Easy access and link still works
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	Unclear what the dashboard is about at first glance	Unsure where to click first, no intro guide	Confused by technical terms like "fiscal health"	No way to save session/progress	Forgets link or loses interest if not reminded
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested? Product School	Add a catchy one-line description at the top	Add a short onboarding tooltip or intro popup	Use simple tooltips & visual legends for each indicator	Add quick export option	Add Cpy link or Bookmark button or share via email/ social

3.2 Solution Requirement

Functional Requirements:

These describe what the system should do.

- The system should allow users to upload fruit images.
- The system should preprocess the image (resize, normalize).
- The model should classify the fruit as Fresh or Rotten.
- The system should display prediction results with confidence scores.
- The system should store or log prediction results (if required).
- The system should provide a simple and user-friendly interface.

Non-Functional Requirements:

These describe how the system should perform.

- The system should provide prediction results within a few seconds.
- The model should achieve at least 90% validation accuracy.
- The system should be scalable to support more fruit categories.
- The application should be secure and prevent invalid file uploads.
- The system should be reliable and minimize downtime.
- The interface should be responsive and easy to use.

3.3 Data Flow Diagram

Step-by-Step Data Flow

1. **User Input**
   The user uploads a fruit image through the web interface.

2. **Image Preprocessing**
   - Image is resized (e.g., 224x224).
   - Pixel values are normalized.
   - Image is converted into array format.

3. **Model Processing**
   - The preprocessed image is passed to the trained MobileNetV2 model.
   - The model extracts features and performs classification.

4. **Prediction Generation**
   - Model outputs class label (Fresh/Rotten).
   - Confidence score is generated.

5. **Result Display**
   - Prediction results are displayed on the web application.

Simple Data Flow Diagram (Text Format)

User → Image Upload → Preprocessing → Trained Model → Prediction → Display Result

3.4 Technology Stack

1. Programming Language

Python – Core development, model training, and backend logic.

2. Deep Learning & ML

- **TensorFlow / Keras** – Model building and training.
- **MobileNetV2** – Pre-trained CNN model for transfer learning.
- **Scikit-learn** – Evaluation metrics (accuracy, confusion matrix, classification report).

3. Data Processing

- **NumPy** – Numerical operations.
- **OpenCV / PIL** – Image preprocessing.

4. Web Development

- **Flask** – Backend web framework.
- **HTML, CSS** – Frontend user interface.

5. Visualization

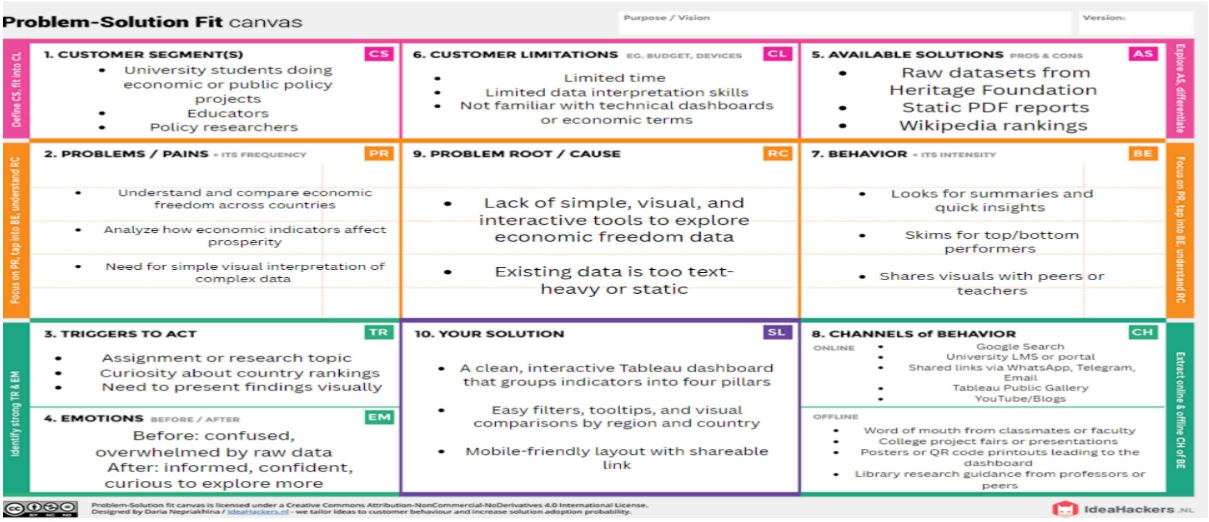
- **Matplotlib / Seaborn** – Training and validation performance graphs.

6. Development Tools

- **Anaconda** – Environment management.
- **VS Code** – Code editor.
- **Jupyter Notebook** – Model experimentation.

4. Project Design

4.1 Problem Statement Fit



Problem-Solution Fit canvas

Purpose / Vision

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

- University students doing economic or public policy projects
- Educators
- Policy researchers

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- Limited time
- Limited data interpretation skills
- Not familiar with technical dashboards or economic terms

5. AVAILABLE SOLUTIONS PROS & CONS AS

- Raw datasets from Heritage Foundation
- Static PDF reports
- Wikipedia rankings

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- Understand and compare economic freedom across countries
- Analyze how economic indicators affect prosperity
- Need for simple visual interpretation of complex data

9. PROBLEM ROOT / CAUSE RC

- Lack of simple, visual, and interactive tools to explore economic freedom data
- Existing data is too text-heavy or static

7. BEHAVIOR + ITS INTENSITY BE

- Looks for summaries and quick insights
- Skims for top/bottom performers
- Shares visuals with peers or teachers

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

- Assignment or research topic
- Curiosity about country rankings
- Need to present findings visually

4. EMOTIONS BEFORE / AFTER EM

Before: confused, overwhelmed by raw data
After: informed, confident, curious to explore more

10. YOUR SOLUTION SL

- A clean, interactive Tableau dashboard that groups indicators into four pillars
- Easy filters, tooltips, and visual comparisons by region and country
- Mobile-friendly layout with shareable link

8. CHANNELS of BEHAVIOR CH

ONLINE

- Google Search
- University LMS or portal
- Shared links via WhatsApp, Telegram, Email
- Tableau Public Gallery
- YouTube/Blogs

OFFLINE

- Word of mouth from classmates or faculty
- College project fairs or presentations
- Posters or QR code printouts leading to the dashboard
- Library research guidance from professors or peers

Extract online & offline CH of BE

IdeaHackers.NL

4.2 Proposed Solution

The proposed solution is to develop a **deep learning-based image classification system** that automatically identifies whether a fruit is fresh or rotten.

The system uses **transfer learning with MobileNetV2**, where a pre-trained convolutional neural network is fine-tuned on a fruit image dataset. Users upload an image through a web interface, the image is preprocessed (resized and normalized), and the trained model predicts the class with a confidence score.

The solution is deployed as a **Flask-based web application**, providing fast, accurate, and real-time freshness detection while reducing manual inspection effort and human error.

4.3 Solution Architecture

The solution architecture of the Fruit Freshness Detection System follows a three-layer structure consisting of the presentation layer, application layer, and model layer. The presentation layer is built using HTML and CSS, allowing users to upload fruit images and view prediction results. The application layer is developed using Flask in Python, which handles image uploads, performs preprocessing such as resizing and normalization, and communicates with the trained deep learning model. The model layer uses TensorFlow/Keras with MobileNetV2 (transfer learning) to extract features and classify the fruit as fresh or rotten, generating a confidence score. The prediction is then sent back to the frontend and displayed to the user in real time.

5.Project Planning Phase

5.1 Project Planning

The project was planned in a structured and phased manner to ensure smooth development and timely completion. Initially, the problem was identified and requirements were gathered. In the next phase, the fruit image dataset was collected and preprocessed. Model development was carried out using MobileNetV2 with transfer learning, followed by training and evaluation using appropriate performance metrics. After achieving satisfactory accuracy, fine-tuning was performed to improve validation results. In the final phase, the trained model was integrated into a Flask-based web application, tested thoroughly, and prepared for deployment. Regular testing and validation were conducted at each stage to ensure reliability and performance.

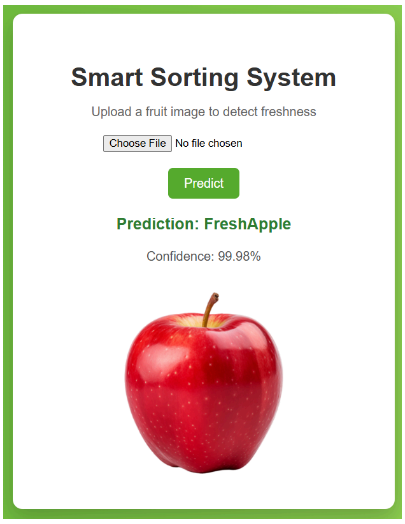
6.Functional and Performance Testing

6.1 Performance Testing

Performance testing was conducted to evaluate the accuracy, reliability, and efficiency of the Fruit Freshness Detection System. The model was tested using separate training and validation datasets to measure its generalization capability. Key performance metrics such as accuracy, confusion matrix, precision, recall, F1-score, and loss values were analyzed. The system achieved high training and validation accuracy with minimal overfitting. Fine-tuning was performed to further improve validation performance. Additionally, response time was tested to ensure that the system provides predictions within a few seconds after image upload, confirming its suitability for real-time usage.

7. Results

7.1 Output Screenshots

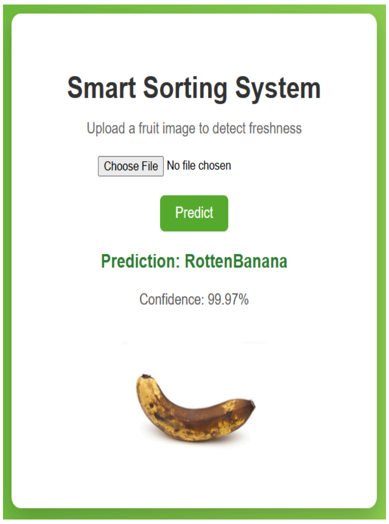


Smart Sorting System

Upload a fruit image to detect freshness

Prediction: FreshApple

Confidence: 99.98%

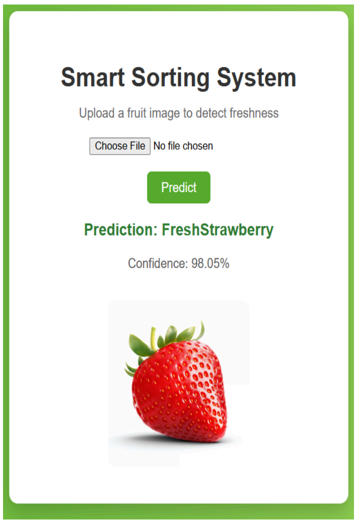


Smart Sorting System

Upload a fruit image to detect freshness

Prediction: RottenBanana

Confidence: 99.97%



Smart Sorting System

Upload a fruit image to detect freshness

Prediction: FreshStrawberry

Confidence: 98.05%

8. Advantages & Disadvantages

Advantages

- Provides **high accuracy** using deep learning and transfer learning techniques.
- Reduces **manual inspection effort** and human error.
- Delivers **fast and real-time predictions**.
- User-friendly web interface for easy image upload and result display.
- Scalable – can be extended to support more fruit categories.
- Improves quality control in agriculture and food industries.

Disadvantages

- Requires a **large and well-labeled dataset** for better accuracy.
- Performance depends on **image quality and lighting conditions**.
- May not perform well on unseen fruit varieties.
- Requires computational resources for model training.
- Internet/server dependency if deployed online.

9.Conclusion

The Fruit Freshness Detection System successfully demonstrates how deep learning and transfer learning can be used to automate fruit quality classification. By using MobileNetV2 and a Flask-based web application, the system accurately identifies whether a fruit is fresh or rotten with high validation accuracy. The solution reduces manual effort, minimizes human error, and provides fast, real-time predictions. Overall, the project proves that AI-based image classification can be effectively applied in agriculture and food quality management systems, with potential for further expansion and improvement.

10. Future Scope

The project can be extended to support a wider variety of fruits and vegetables to make the system more comprehensive. The model can be further improved by training on a larger and more diverse dataset to enhance accuracy under different lighting and background conditions. In the future, the system can be integrated with mobile applications for on-field usage by farmers and vendors. It can also be connected with IoT devices or smart sorting machines for automated fruit grading in industries. Additionally, features like multi-class classification (fresh, semi-fresh, rotten) and real-time camera-based detection can be implemented to make the solution more advanced and practical.

11. Appendix

Dataset Link:

https://www.kaggle.com/datasets/abdulrafeyyashir/fresh-vs-rotten-fruit-images?select=Fruit+Freshness+Dataset

Github Link:

https://github.com/Ak26-9/Smart-Sorting-Transfer-learning-for-identifying-rotten-fruits-and-vegetables