**Brainstorm & Idea Priorotization**

| **Field** | **Details** |
| --- | --- |
| **Date** | 16-06-2025 |
| **Team ID** | LTVIP2025TMID35102 |
| **Project Name** | Smart Sorting: Detecting Rotten Fruits with Transfer Learning and Flask Integration |
| **Maximum Marks** | 4 Marks |

**Objective:**

To develop a smart image classification system that helps in identifying rotten fruits or vegetables using deep learning and transfer learning techniques. This project promotes reduced food waste and enhances food safety, especially for retailers, vendors, and storage hubs.

**Step 1: Team Gathering, Collaboration and Problem Identification**

The team collaborated to understand challenges faced in the agri-retail and food logistics industry, particularly around the early identification of rotten produce. We discussed the practical implications of rot detection in warehouses, supply chains, supermarkets, and home storage.

**Real-world issues identified:**

* Manual sorting is time-consuming and prone to error.
* Lack of affordable, quick detection tools for rotten produce.
* Visual rot often escapes human detection in bulk storage.
* Spoiled produce leads to economic losses and health hazards.
* Farmers and small vendors lack automation in quality check.

**Selected Problem Statement:**  
There is a need for a low-cost, automated, and accurate system to detect rotten fruits and vegetables using image classification and deep learning.

**Step 2: Brainstorming, Idea Listing and Grouping**

**Raw Ideas Collected:**

* Use CNN with transfer learning (e.g., VGG16)
* Classify images as Fresh vs Rotten
* Build user-friendly Flask web app
* Allow image uploads via browser/mobile
* Show prediction results visually with confidence score
* Store results in local log or database
* Create information section on food storage tips
* Add camera support for real-time detection
* Display example images after prediction
* Deploy on Render or Railway for demo

**Grouped Ideas:**

**1. Model Development**

* Use pre-trained VGG16 and fine-tune for fruit rot classification
* Augment data for robustness
* Export .h5 model for Flask integration

**2. Web App Integration**

* Flask-based frontend with HTML/CSS
* Upload button and prediction display
* User-friendly UI/UX

**3. Feature Extensions**

* Confidence score display
* Image preview before/after upload
* Add example use-cases in UI

**4. Deployment & Hosting**

* Deploy on Render or Railway
* Mobile-friendly access
* Easy to share with mentors or evaluators

**5. Future Scope**

* Real-time camera prediction
* Expand dataset to detect mold stages
* Suggest best-before dates or shelf life estimation

**Step 3: Idea Prioritization**

| **Idea** | **Impact** | **Feasibility** | **Priority** |
| --- | --- | --- | --- |
| **Rotten vs Fresh classification** | **High** | **High** | **High** |
| **Flask web interface** | **High** | **Medium** | **High** |
| **Image upload + preview** | **High** | **High** | **High** |
| **Deploy online (Render/Railway)** | **Medium** | **Medium** | **Medium** |
| **Camera-based prediction** | **High** | **Low** | **Low** |
| **Info section (storage tips)** | **Medium** | **Medium** | **Medium** |
| **Display confidence score** | **High** | **High** | **High** |

**Key Functionalities (Examples):**

**1. Fresh vs Rotten Prediction Module**

Functionality: Upload image → Get classification result  
Technology Used: VGG16, Keras, Flask  
Benefits: Fast and accurate detection for users

**2. Flask UI with Image Upload**

Functionality: Upload image from browser or phone, and get prediction  
Technology Used: Flask, HTML, CSS  
Benefits: Makes the model accessible to all users

**3. Confidence Score Display**

Functionality: Show probability of prediction (e.g., 94% Fresh)  
Technology Used: Keras model output + conditional formatting  
Benefits: Builds user trust with transparency

**4. Example Use Cases Section**

Functionality: Static UI section with cards explaining where this system can be used  
Technology Used: HTML templates  
Benefits: Highlights real-world relevance of the project

**5. Cloud Deployment**

Functionality: Make the web app available via public URL  
Technology Used: Render, Railway  
Benefits: Easy sharing for demo, mentor access, and evaluations