SMART SORTING PROJECT

A Classification of fruits and vegetables whether they are fresh or Rotten

SRI VENKATESWARA COLLEGE OF ENGINEERING

Team ID : LTVIP2025TMID41462

Team Size : 5

Team Leader : Chamarthi Lasya

Team member : D Dharani

Team member : Gadde Venkata Sruthi

Team member : Pallamparthi Sonipriya

Team member : Gandi Bhuvaneswari Devi

Ideation Phase
Brainstorm & Idea Prioritization Template

Date	30 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Objective:

The objective of this project is to develop an intelligent image classification system that can accurately categorize fruits and vegetables into four distinct classes — fresh fruit, fresh vegetable, rotten fruit, and rotten vegetable — using computer vision and deep learning techniques.

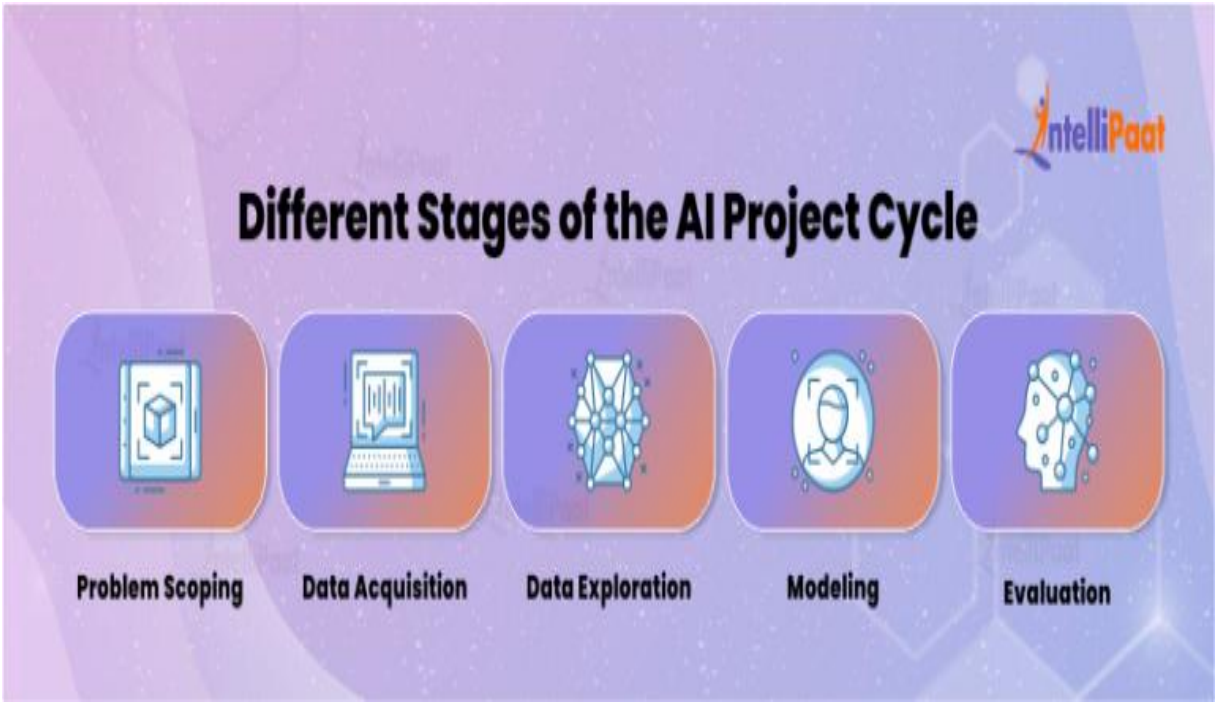
Step-1: Team Gathering, Collaboration and Select the Problem Statement

The team conducted an initial discussion session to identify key real-world problems where Artificial Intelligence can bring impactful solutions. We explored various areas including agriculture, food safety, and logistics.

After a shortlisting phase, we selected the problem statement:

"Develop a smart system that can classify fruits and vegetables based on their freshness using computer vision."

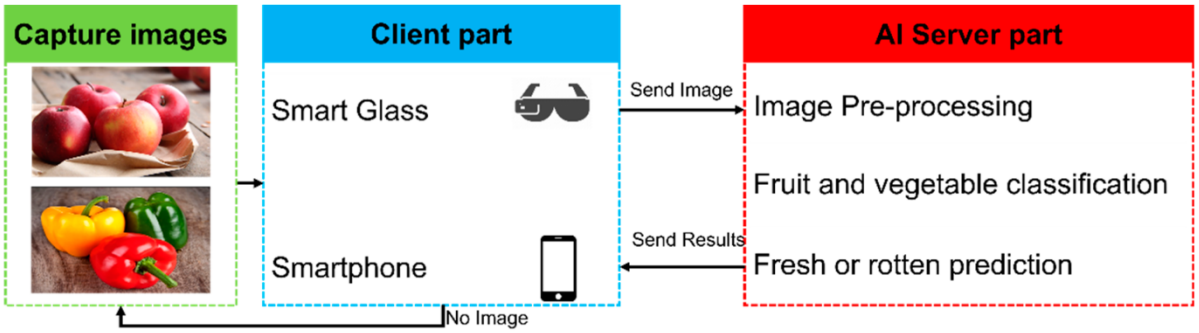
This problem is critical in reducing food waste, optimizing food sorting in markets, and ensuring better food quality for consumers.



Step-2: Brainstorm, Idea Listing and Grouping

We brainstormed various ideas and grouped them under the following categories:

- Classify fresh/stale fruits using image processing
- Detect mold or discoloration
- Build a web app with Flask
- Use MobileNetV2 for fast, lightweight model
- Host model as `.h5` and load dynamically



Step-3: Idea Prioritization

We used the Impact vs. Feasibility Matrix to score each idea:

Idea	Impact	Feasibility	Priority
Classify fresh/rotten produce using Mobile Net	High	High	Selected
Integrate IOT hardware	Medium	Low	Postponed
Use CNN from scratch	Medium	Low	Not Selected
Deploy as Android App	High	Medium	Future Scope

Final Prioritized Idea:

Smart Sorting Web App that classifies produce into 4 categories using a trained CNN model.

Ideation Phase
Define the Problem Statements

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	2 Marks

Customer Problem Statement Template:

As a farmer/vendor/warehouse worker, I face difficulties in quickly and accurately identifying whether fruits and vegetables are fresh or rotten. Manual inspection is time-consuming, inconsistent, and often leads to spoilage, customer complaints, and financial losses. I need a fast, reliable, and easy-to-use solution that can help me sort produce based on quality, without requiring technical knowledge or manual effort.

I am	I'm trying to	But	Because	Which makes me feel
A homemaker	Buy only fresh fruits and vegetables	Sometimes I get stale or rotten items from the market	It’s hard to identify the freshness visually	Frustrated and disappointed with my purchases
A vendor selling groceries	Sell only fresh items to my customers	I can’t quickly check freshness of large quantities	Manual checking is time-consuming and unreliable	Worried about customer trust and losing business
A restaurant chef	Ensure all cooking ingredients are top quality	I receive spoiled or stale produce occasionally	My suppliers don't always inform me of freshness properly	Concerned about customer health and reputation
A student living in a hostel	Buy healthy food on a tight budget	I often end up buying low-quality vegetables from the local store	I lack knowledge and tools to assess freshness	Dissatisfied and confused when selecting produce
A health-conscious customer	Eat only nutritious and hygienic food	I can't differentiate rotten from fresh visually	Supermarkets don’t label the freshness of loose produce	Anxious about food quality and health

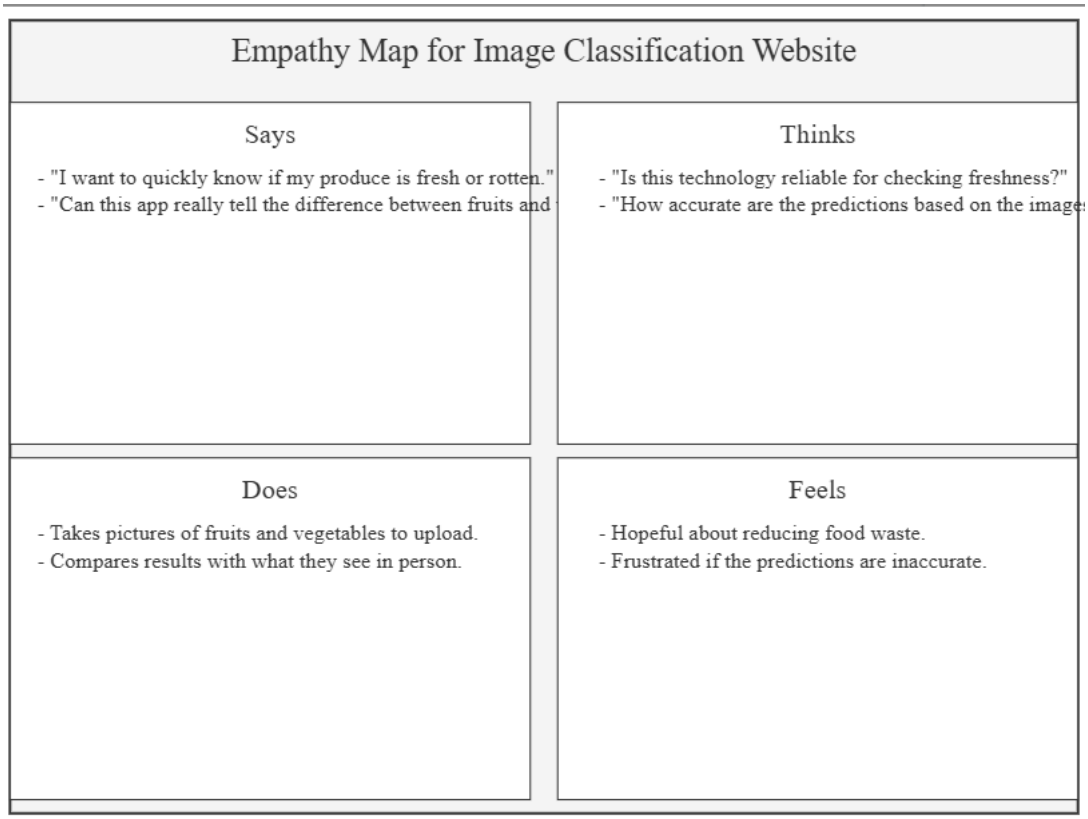
Ideation Phase
Empathize & Discover

Date	30 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	4 Marks

Empathy Map:

This Empathy Map is created considering the customer or quality inspector as the user**.** The empathy map will include organized text to clearly represent the users' thoughts, feelings, actions, and statements.

Example:



Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

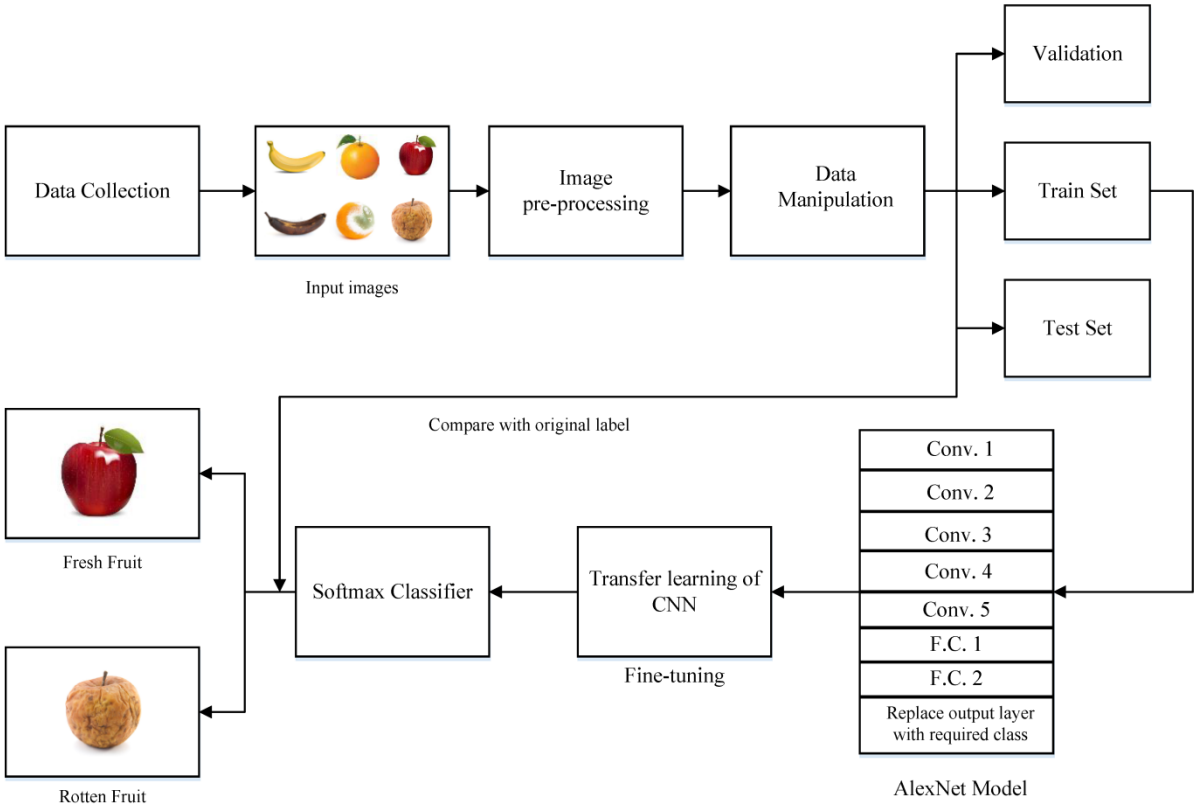
Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1 .	Image Upload Interface	User can upload fruit or vegetable images through a clean, responsive UI
FR-2	Image Pre processing	Automatically resizes and scales input images to the required dimensions (224×224).
FR-3	Classification Model Integration	Predicts one of 4 classes: fresh fruit, fresh vegetable, rotten fruit, rotten vegetable.
FR-4	Result Display	Displays the predicted category along with the uploaded image for user confirmation.
FR-5	Dataset Handling	Balances training and testing dataset from 12 folders into 4 target classes with equal distribution.
FR-6	File Upload & Backend Connectivity	Uses Flask to handle image uploads and run model inference seamlessly in the background.
FR-7	Model Storage & Loading	Saves trained model as `.h5` file and reloads it during real-time prediction.

Non-Functional Requirements:

NFR No.	Non-Functional Requirement	Description
NFR-1	Accuracy	Target model accuracy between 90% to 95% to ensure reliable predictions.
NFR-2	Performance	Prediction response time must be under 2 seconds for a seamless user experience
NFR-3	Scalability	Should support future addition of more classes (e.g., leaf diseases, other produce).
NFR-4	Usability	Intuitive interface accessible to both farmers and sellers with no technical background.
NFR-5	Portability	Flask app must run on any system with Python & TensorFlow installed (cross-platform).
NFR-6	Accessibility	Web app should display well on laptops, tablets, and smartphones (mobile responsive).
NFR-7	Data Integrity	Ensures correct class mapping from 12 folders into 4 classes; no misclassification in datasets.

DATA FLOW DIAGRAM



Technology Stack

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Technology Stack:

Category	Technology / Tool	Purpose
Programming Language	Python	Core programming for model training and Flask backend
Deep Learning Framework	TensorFlow / Keras	Building and training the CNN classification model
Pre-trained Model	MobileNetV2	Transfer learning for fast and accurate image classification
Data Handling	NumPy, Pandas	Data preprocessing and handling
Image Processing	Pillow, OpenCV (optional)	Reading and manipulating images
Web Framework	Flask	Backend web server to serve predictions
Frontend	HTML5, CSS3	User interface for uploading images
Dataset Source	Kaggle Dataset: Fresh and Stale Fruits & Vegetables	Training and testing the classification model
Deployment Platform (optional)	PythonAnywhere / Render / Localhost	Hosting the Flask web app
Model Format	`.h5` (HDF5)	Saving the trained Keras model
IDE / Notebook	Google Colab	Training and evaluating the model in a cloud environment
Version Control	Git + GitHub	Project management and submission
Visualization	Matplotlib	Optional: Plotting training and validation accuracy/loss

Customer Journey Map

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Customer Journey Map

Customer Persona: Retail Quality Inspector / Vendor / Household User
Product: Smart Sorting Web App using AI (Fresh/Rotten Fruit & Vegetable Classifier)
Goal: Accurately detect and classify produce quality through image upload to reduce wastage and ensure freshness.

Overview Table

Here is the content in a table format:

Stage	Touch points	Customer Actions	Experience	Pain Points	Opportunities for Improvement
Awareness	Word-of-mouth, Social media, Posters	Learns about the Smart Sorting app	Curious	Unsure if tool is accurate or works on phone	Provide demo video, feature list on landing page
Consideration	Smart Sorting Homepage	Explores how to use app, uploads a test image	Cautiously interested	UI slightly confusing for first-time user	Add tooltips or welcome guide popup
On -boarding	File Upload UI, Prediction Result page	Uploads produce image, views prediction	Impressed	Upload size limit or format mismatch	Allow drag-drop, add supported formats info
Usage	Flask App Backend, Trained Model	Uploads bulk images daily to sort produce	Productive, efficient	Wants multi-image upload or mobile app	Add batch upload feature, Android version (future scope)
Monitoring	Prediction Result Screen, Confidence	Uses results to separate produce for sale	Confident	Wants confidence score shown clearly	Add bar meter or percentage below prediction label

Maintenance	Image retraining (future feature)	Feeds new images to improve model (v2)	Engaged	Needs feedback option	Add “Did we get it right?” button
Support	FAQ, Chatbot (future), Email	Asks about low accuracy or image issues	Supported	Limited immediate help	Include chatbot / help widget
Feedback	Feedback Form, GitHub Issues	Reports wrong classifications or suggests improvements	Cooperative	No clear follow-up on suggestions	Add feedback status tracker or thank you email

Project Design Phase
Problem – Solution Fit

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Customer Problem Identified

Consumers and vendors often struggle to manually identify whether fruits or vegetables are fresh or spoiled, leading to wastage, economic losses, and health hazards.

Observed Behavior

People rely on visual inspection or touch, which is inconsistent, time-consuming, and often unreliable — especially in markets and grocery stores.

Proposed Solution

An AI-powered image classification tool that allows users to upload a fruit or vegetable image and instantly get a prediction on its quality: Fresh or Rotten, and whether it is a Fruit or Vegetable.

Why This Solution Works

The model is trained on a balanced dataset of over 2,500 high-quality images across 4 categories and uses MobileNetV2 with 72%+ validation accuracy. This solution provides fast, consistent, and reliable results that eliminate human error.

Unique Value Proposition

Enables quick, mobile-compatible detection with minimal user interaction; helps reduce waste, improve purchase decisions, and support smart inventory sorting in markets and grocery chains.

Impact of the Solution

Saves time, reduces health risk from spoiled items, improves trust in vendors, and encourages better food quality practices. Can be integrated into mobile/web platforms or retail POS systems.

Project Design Phase
Proposed Solution

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Proposed Solution:

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Customers, vendors, and retailers often struggle to manually identify whether fruits and vegetables are fresh or rotten, leading to wastage, poor quality assurance, and health concerns.
2	Idea / Solution Description	An AI-based web application using a deep learning model that classifies uploaded images of fruits and vegetables into four categories: fresh fruit, rotten fruit, fresh vegetable, and rotten vegetable.
3	Novelty / Uniqueness	Uses a custom-trained deep learning model (MobileNetV2) fine-tuned with a 4-class balanced dataset and integrated into an interactive Flask-based website to provide real-time image predictions.

4	Social Impact / Customer Satisfaction	Reduces food wastage, improves consumer trust in produce quality, assists vendors in sorting stock, and empowers users with a quick, accessible tool for quality checking at home or markets.
5	Business Model (Revenue Model)	Free for consumers; potential monetization via API licensing for retailers, quality control agencies, or Agritech companies; also suitable for integration in smart market systems.
6	Scalability of the Solution	Can be scaled to include more fruits/vegetables, support mobile apps, and deployed in grocery stores, warehouses, or smart kiosks. Multilingual support and offline prediction possible for wider adoption.

Solution Architecture

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

Smart Sorting Solution Architecture

1. User Interface (Frontend)

- **Technology:** HTML5, CSS3 (Bootstrap or Custom CSS)
- **Functionality:**
  - User uploads an image (fruit or vegetable)
  - Displays prediction result with image preview
  - Displays class (fresh_fruit, fresh_vegetable, rotten_fruit, rotten_vegetable)

2. Flask Web Server (Backend)

- **Technology:** Python (Flask)
- **Functionality:**
  - Handles user requests (`/` for upload form, `/predict` for classification)
  - Loads `.h5` trained model
  - Preprocesses uploaded image
  - Sends image to model and returns prediction
  - Renders output to frontend with prediction class and image preview

3. Machine Learning Model

- **Framework:** TensorFlow Keras
- **Model:** MobileNetV2 (with transfer learning)
- **Output:** 4-class classification (`softmax`)
- **Accuracy:** Trained to ~72% validation accuracy

4. Data Pipeline

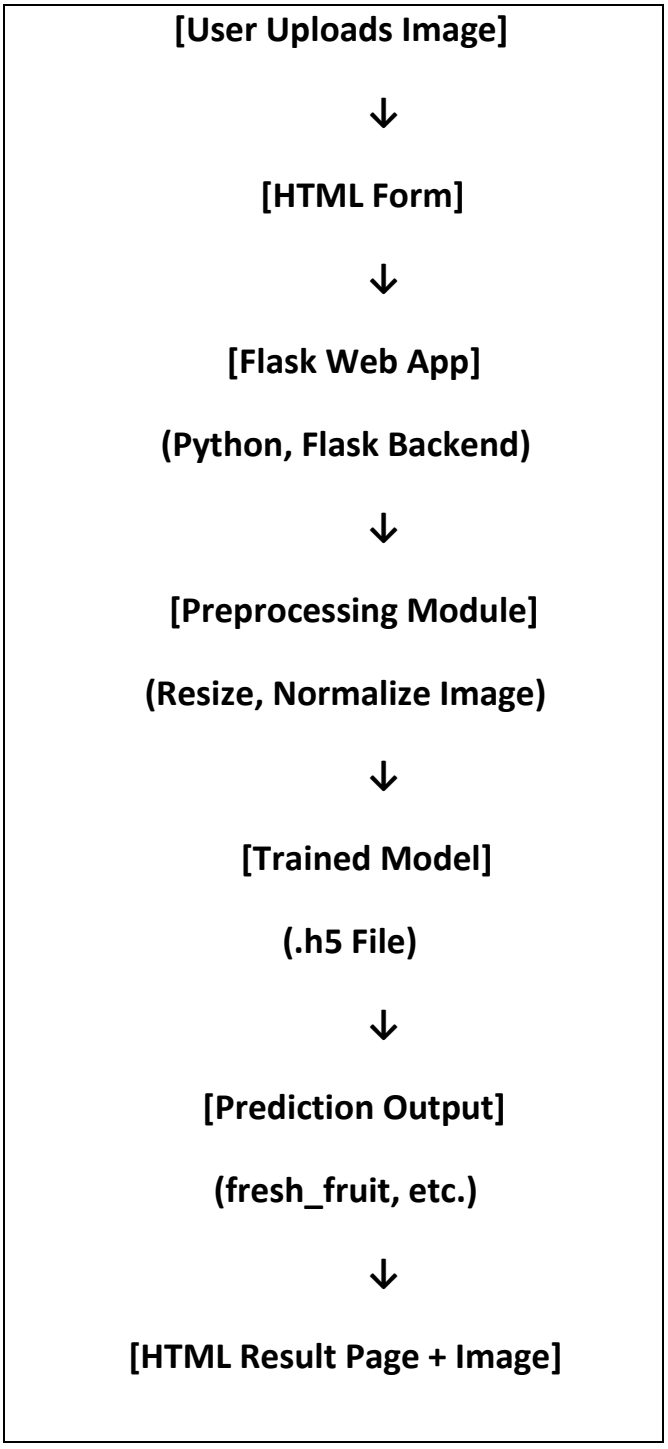
- **Dataset:** Kaggle dataset with 12 folders, mapped into 4 custom classes
- **Preprocessing:**
  - Resized to 224x224
  - Augmentation (rotation, zoom, flip)
  - Train/Validation split
- **Storage:** Google Drive for `.h5` model, `data_subset_4class` for training/testing

5. Deployment (Local / Cloud)

- **Option 1 (Local):**
  - Run Flask app locally on Anaconda / Python virtual environment

- Open `http://127.0.0.1:5000/` in browser
- **Option 2 (Cloud):**
  - Host on platforms like:
    - **Render**
    - **Replit**
    - **Heroku** (with some modifications)
  - (Optional) Integrate with Flask-Ngrok for public URL during development

Architecture Diagram:



Project Planning Phase

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

1. Project Description

Smart Sorting AI is an image classification system developed using deep learning that classifies input images of fruits and vegetables into four categories:

- Fresh Fruit
- Rotten Fruit
- Fresh Vegetable
- Rotten Vegetable

The model uses MobileNetV2 and a balanced dataset to achieve ~72% accuracy, with future scope to improve it further.

2. Project Objectives

- Create a working deep learning model for smart sorting.
- Design a web interface using Flask for real-time image predictions.
- Enable users to upload images and get class predictions.
- Train a model on a 4-class labeled dataset from Kaggle.
- Make the solution lightweight and deployable locally.

3. Technology Stack

Layer	Tools/Frameworks Used
Programming	Python 3.10
Deep Learning	TensorFlow, Keras
Model Architecture	MobileNetV2 (Pre-trained on ImageNet)
Dataset	Kaggle Dataset (Fresh & Stale Fruits/Vegetables)
Web Framework	Flask (Python microframework)
Frontend	HTML, CSS (custom styled form)
Image Handling	Pillow (PIL), base64 for rendering images
Hosting (optional)	Localhost / Flask-Ngrok (for demo)

---

4. Resources Required

Resource Type	Description
Hardware	Google Colab (GPU), Local PC with Anaconda

Resource Type	Description
Dataset	Kaggle: Fresh and Stale Images Dataset
IDEs/Editors	Google Colab, VS Code, Notepad++
Libraries	TensorFlow, Pillow, Flask, Flask-Ngrok
Collaboration Tools	Google Drive, GitHub, PDF editor

---

5. Risk Analysis & Mitigation

Risk	Mitigation Strategy
Overfitting due to small subset	Use dropout, image augmentation
Accuracy too low (<70%)	Tune layers, try more data, longer training
Deployment errors (Flask 404, etc.)	Test each route locally, add validations
File compatibility (image types, .h5)	Standardize formats and handle exceptions

---

6. Success Criteria

- Accuracy ≥ 70% on validation set
- Smooth prediction interface (image → result)
- Working website with upload and result display
- Fully documented GitHub repo and demo video

Project Development Phase
Model Performance Testing

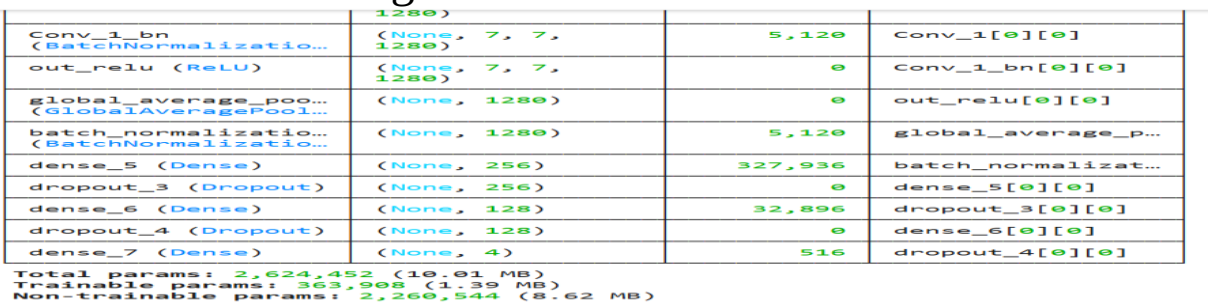
Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables

Model Performance Testing:

S.NO.	Parameter	Values
1.	Model Summary	MobileNetV2 (Transfer Learning), with added layers: - GlobalAveragePooling2D - Batch Normalization - Dense(256, Relu) - Dropout(0.4) - Dense(128, Relu) - Dropout(0.3) - Output: Dense(4, soft max)
2.	Accuracy	Training Accuracy – 75% Validation Accuracy -72%
3.	Fine Tuning Result	Validation Accuracy -Validation Accuracy after tuning: *72.25%* (same as base)

Screenshots:

The screenshots are given below



	1280)		
Conv_1_bn (BatchNormalizatio…	(None, 7, 7, 1280)	5,120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_poo… (GlobalAveragePool…	(None, 1280)	0	out_relu[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 1280)	5,120	global_average_p…
dense_5 (Dense)	(None, 256)	327,936	batch_normalizat…
dropout_3 (Dropout)	(None, 256)	0	dense_5[0][0]
dense_6 (Dense)	(None, 128)	32,896	dropout_3[0][0]
dropout_4 (Dropout)	(None, 128)	0	dense_6[0][0]
dense_7 (Dense)	(None, 4)	516	dropout_4[0][0]

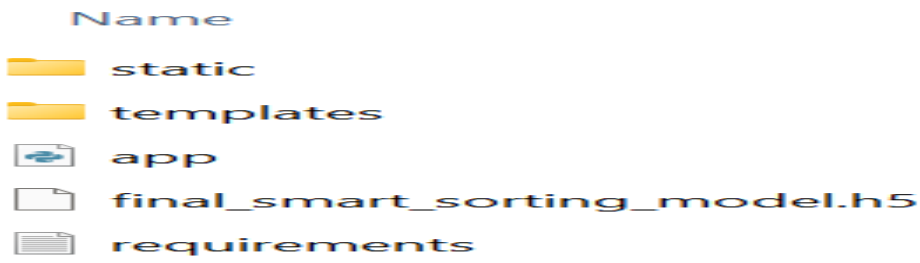
Total params: 2,624,452 (10.01 MB)
Trainable params: 363,908 (1.39 MB)
Non-trainable params: 2,260,544 (8.62 MB)

```
Epoch 25/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9793 - loss: 0.0425
Epoch 25: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 142s 2s/step - accuracy: 0.9794 - loss: 0.0424 - val_accuracy: 0.7225 - val_loss: 1.9970
Epoch 26/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9897 - loss: 0.0294
Epoch 26: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 109s 2s/step - accuracy: 0.9897 - loss: 0.0295 - val_accuracy: 0.7225 - val_loss: 2.1223
Epoch 27/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9867 - loss: 0.0333
Epoch 27: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 140s 2s/step - accuracy: 0.9868 - loss: 0.0332 - val_accuracy: 0.7200 - val_loss: 2.0702
Epoch 28/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9843 - loss: 0.0395
Epoch 28: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 109s 2s/step - accuracy: 0.9844 - loss: 0.0394 - val_accuracy: 0.7150 - val_loss: 2.1686
Epoch 29/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9887 - loss: 0.0390
Epoch 29: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 128s 3s/step - accuracy: 0.9888 - loss: 0.0389 - val_accuracy: 0.7125 - val_loss: 2.2191
Epoch 30/30
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9900 - loss: 0.0355
Epoch 30: val_accuracy did not improve from 0.73250
51/51 ━━━━━━━━━━━━━━━━━━━━ 108s 2s/step - accuracy: 0.9900 - loss: 0.0355 - val_accuracy: 0.6950 - val_loss: 2.1674
```

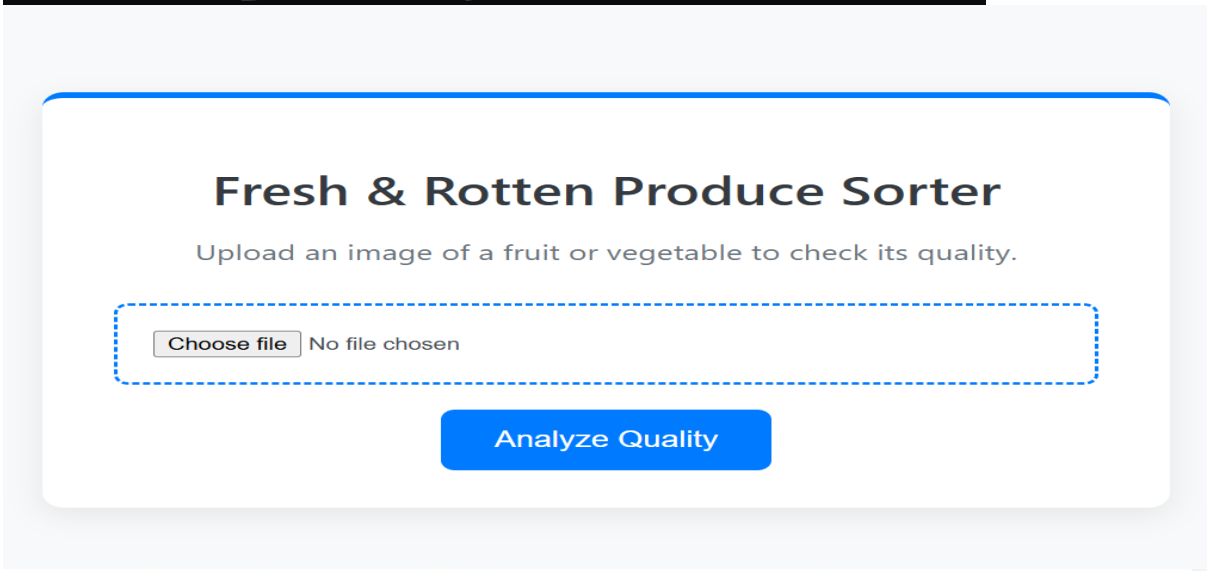
```
[ ] best_model = tf.keras.models.load_model(os.path.join(PROJECT_PATH, 'healthy_vs_rotten_4class.h5'))
    loss, acc = best_model.evaluate(val_gen)
    print(f"Final Validation Accuracy: {acc * 100:.2f}%")
```

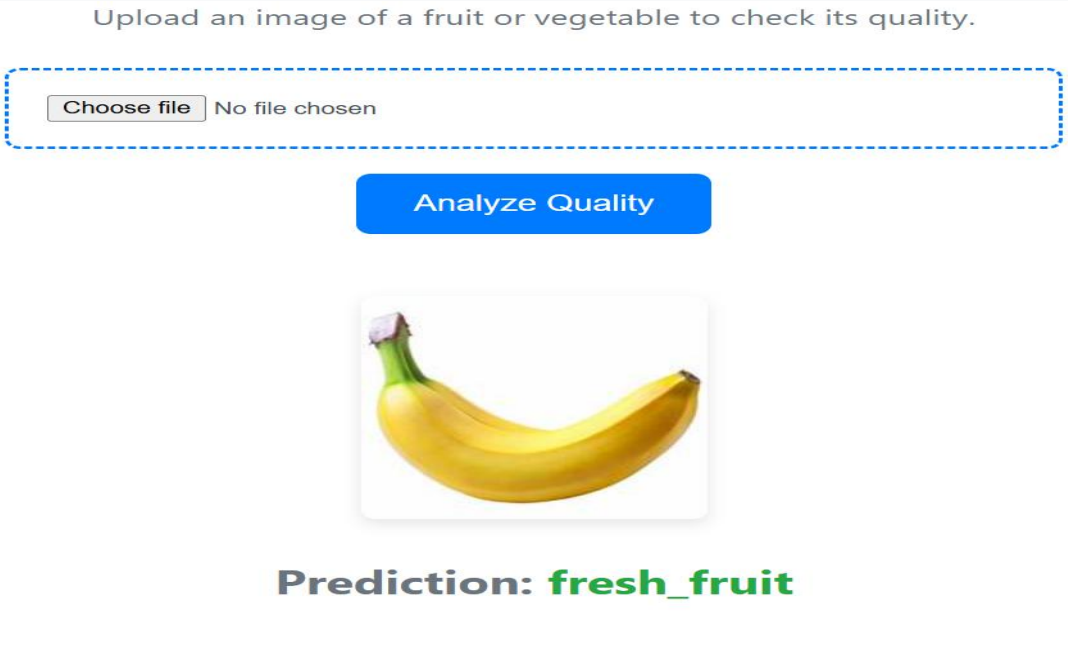
```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_me
13/13 ━━━━━━━━━━━━━━━━━━━━ 26s 2s/step - accuracy: 0.7124 - loss: 1.7133
Final Validation Accuracy: 72.25%
```

```
# STEP 10: Manually save the model
model.save(os.path.join(PROJECT_PATH, 'final_smart_sorting_model.h5'))
print("✅ Model saved as final_smart_sorting_model.h5")
```




```
* Running on http://127.0.0.1:5000
```





Prediction: rotten_fruit

Prediction: rotten_vegetable

Upload an image of a fruit or vegetable to check its quality.

Choose file No file chosen

Analyze Quality

Prediction: fresh_vegetable