

## Project Design Phase

### Problem – Solution Fit Template

Date	19 February 2026
Team ID	LTVIP2026TMIDS77295
Project Name	Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables.

## Problem–Solution Phase

### 1. Nature of the Problem

The core problem is to classify images of fruits and vegetables into two categories: Fresh or Rotten. This is a supervised image classification problem with several challenges:

- Spoilage appears in many subtle forms (small brown spots, color fading, mold) that are hard to capture with simple rules.
- Images vary in lighting, angle, background, and camera quality.
- The system must work fast enough to be usable in real environments like markets and warehouses.

Because of this, traditional rule-based or classical machine learning methods (e.g., color thresholds, hand-crafted features + SVM) are not robust enough; they struggle when the lighting or viewpoint changes or when multiple defects appear together.

---

### 2. Chosen Approach: Deep Learning with Transfer Learning (MobileNetV2)

To handle the complexity of visual patterns in rotten vs fresh produce, the project uses a Convolutional Neural Network (CNN) with transfer learning, specifically MobileNetV2 as the base model.

Key idea: Instead of designing features by hand, a CNN automatically learns features like edges, textures, color patterns, and shapes directly from images. Transfer learning reuses a network already trained on a large dataset (ImageNet) and adapts it to our Fresh/Rotten classification task.

---

### 3. Why We Selected MobileNetV2 (Algorithm Justification)

#### a) High accuracy on visual tasks

Pre-trained CNNs like MobileNetV2 have already learned rich visual features from millions of images.

Studies on fruit quality and freshness detection show that transfer learning with MobileNet-type models reaches high accuracy for fresh vs rotten classification.

This means we can achieve strong performance even with a limited domain dataset.

#### b) Works well with limited dataset size

Collecting thousands of labeled fruit images for every condition is difficult.

Training a CNN from scratch would require huge data and high compute, and would likely overfit. MobileNetV2 transfer learning needs fewer training samples because most of the low-level filters are reused; we only train the top layers for the new binary task.

#### c) Lightweight and fast for real-time use

Compared to large networks (e.g., ResNet-152, Inception-ResNet), MobileNetV2 is designed to be lightweight and efficient, originally for mobile and embedded devices.

This is important because:

- Our application should run on normal CPUs (college lab systems, small servers, or low-cost machines).
- Users expect quick response when they upload an image—near real-time classification.

Thus, MobileNetV2 gives a good trade-off between speed and accuracy for deployment in small and medium-scale environments.

#### **d) Easier integration with Flask and TensorFlow**

MobileNetV2 is available directly in TensorFlow/Keras applications with a single function call, including pre-trained weights and preprocessing utilities.

This simplifies:

- Loading the model in train.py and predict.py.
- Saving and re-using weights (.h5 file) in the Flask backend.
- Maintaining a clean and modular codebase.

Because Flask and TensorFlow are both Python-based, there is no language barrier in integration.

#### **e) Proven success in similar domains**

Research and open-source projects have successfully used CNNs and transfer learning (MobileNet, VGG, etc.) for fruit quality detection, ripeness prediction, and defect detection.

Adopting a similar architecture is a safe and evidence-based engineering choice.

---

## **4. How the Algorithm Solves the Problem**

### **1. Feature Extraction**

- MobileNetV2 processes the input image through multiple convolutional layers and depthwise separable convolutions.
- These layers learn hierarchical features: edges → textures → shapes → high-level patterns related to spoilage.

### **2. Transfer Learning and Fine-Tuning**

- The pre-trained base is initially frozen; a new dense head is trained on our dataset to distinguish Fresh vs Rotten.
- Optionally, upper layers of MobileNetV2 are unfrozen and fine-tuned with a smaller learning rate to better adapt to fruit images.

### **3. Binary Classification**

- The final layer outputs a probability between 0 and 1 (or two softmax probabilities) indicating whether the fruit is fresh or rotten.
- A threshold (e.g., 0.5) is used to choose the class; the probability is shown to the user as a confidence score.

### **4. Robustness to Real-World Variation**

- Data augmentation (rotations, flips, brightness changes) combined with MobileNetV2's learned features makes the model robust to changes in angle, lighting, and background.
- This is crucial for farm or market images, which are rarely perfect.

## 5. Why We Did Not Choose Other Methods

- Manual rule-based methods (color thresholding, simple morphology)
  - Highly sensitive to lighting and camera variations.
  - Hard to generalize to many fruit types and subtle defects.
  - Would require continuous manual tuning for each scenario.
- Classical machine learning with hand-crafted features (HOG, color histograms + SVM)
  - Requires feature engineering for each new fruit type or condition.
  - Typically underperforms compared to modern CNNs on complex image tasks.
- Heavier CNNs (e.g., ResNet-50/101, Inception-ResNet)
  - Might give slightly higher accuracy but are slower and more resource-hungry.
  - Not ideal for deployment on typical low-cost systems used by farmers, vendors, or small warehouses.

### Template:

Solution Canvas for 'Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables'						
Driving CS, Before CS Instructing AS Sometime PA Before After	1. CUSTOMER SEGMENT(S)	CS	6. CUSTOMER CONSTRAINTS	CC	5. AVAILABLE SOLUTIONS	AS
	<ul style="list-style-type: none"> <li>• Farmers handling harvest and pre-sorting</li> <li>• Wholesalers / retail fruit &amp; vegetable vendors</li> <li>• Supermarket quality inspectors</li> <li>• Cold-storage and warehouse managers</li> <li>• Food processing units (juice, frozen food, ready-to-eat)</li> </ul>		<ul style="list-style-type: none"> <li>• Detect rotten or near-spoiled fruits and vegetables quickly</li> <li>• Reduce manual inspection time for large quantities</li> <li>• Minimize financial loss due to unnoticed spoilage</li> <li>• Maintain consistent quality standards and brand reputation</li> <li>• Improve customer satisfaction and trust in product freshness</li> </ul>		<ul style="list-style-type: none"> <li>• Limited budget for expensive or inspectors</li> <li>• Low technical expertise in AI/ML &amp; hardware based only</li> <li>• Sometimes poor or unstable internet connectivity</li> <li>• Limited space and power supply in local markets or farms</li> </ul>	
	2. JOBS-TO-BE-DONE / PROBLEMS	J&P	3. TRIGGERS (TR)	(Ds)	7. BEHAVIOUR	BE
	<ul style="list-style-type: none"> <li>• Detect rotten or near-spoiled fruits and vegetables quickly</li> <li>• Reduce manual inspection time for large quantities</li> <li>• Minimize financial loss due to unnoticed spoilage</li> <li>• Maintain consistent quality standards and brand reputation</li> <li>• Improve customer satisfaction and trust in product freshness</li> </ul>		<ul style="list-style-type: none"> <li>• Sudden increase in rotten or returned stock</li> <li>• Customer complaints about bad smell, taste, or appearance</li> <li>• Visible drop in profits due to wastage and discounts on poor stock</li> <li>• Need to process more quantity with the same or fewer staff</li> <li>• Push from management towards digital/AI-based quality control</li> </ul>		<ul style="list-style-type: none"> <li>• Manually pick up fruits/vegetables and visually check defects</li> <li>• Separate visibly rotten pieces into discard baskets</li> <li>• Often detect spoilage only after smell or customer complaint</li> <li>• Rely mainly on experience and intuition instead of data</li> </ul>	
	5. AVAILABLE SOLUTIONS	AS	6. CUSTOMER CONSTRAINTS	CC	9. PROBLEM ROOT CAUSE (RC)	RC
	<ul style="list-style-type: none"> <li>• Manual inspection by workers or inspectors</li> <li>• Basic mechanical sorting machines (size/weight based only)</li> <li>• Traditional visual checking with torch/eye inspection</li> <li>• Occasional lab tests or manual sampling</li> </ul>		<ul style="list-style-type: none"> <li>• Limited budget for expensive hardware or imported machines</li> <li>• Low technical expertise in AI/ML and IT maintenance</li> <li>• Sometimes poor or unstable internet connectivity</li> <li>• Limited space and power supply in local markets or farms</li> </ul>		<ul style="list-style-type: none"> <li>• Quality check depends completely on human vision and attention</li> <li>• Lack of affordable-easy-to-use AI tools for small and mid-scale-taers</li> <li>• No early automated detection of small rot spots or texture changes</li> </ul>	
	4. EMOTIONS (BEFORE / AFTER)	EM	8. CUSTOMER CONSTRAINTS	BE	10. YOUR SOLUTION (SL)	SL
	<p>Before Using Smart Sorting</p> <ul style="list-style-type: none"> <li>• Frustrated with slow and tiring manual checking</li> <li>• Uncertain whether all rotten items are caught in time</li> <li>• Worried about financial losses and negative reviews</li> <li>• Under time pressure during peak hours or seasons</li> </ul>		<ul style="list-style-type: none"> <li>• Agricultural/farming mobile apps and portals</li> <li>• WhatsApp groups of farmers, vendors, FPOs</li> <li>• YouTube channels on farming, storage, and agitech</li> <li>• Online marketplaces for agri-inputs and produce</li> </ul>		<ul style="list-style-type: none"> <li>• Smart Sorting - AI Powered Rotten Produce Detection</li> <li>• Uses Deep Learning with Transfer Learning (MobileNetV2-based CNN) to classify images as Fresh or Rotten.</li> <li>• Simple-web interface: user opens Predict page and uploads a maty ergetable image.</li> <li>• Backend Flask app preprocesses the image and runs the trained model.</li> <li>• System instantly shows prediction label (Fresh/Rotten) with confidence score and proslett of the image.</li> </ul>	
	5. CHANNELS OF BEHAVIOUR (CH)		9. PROBLEM ROOT CAUSE (RC)		10. YOUR SOLUTION (SL)	