Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	18 February 2026
Team ID	LTVIP2026TMIDS80710
Project Name	Smart Sorting:Transfer Learning for Identifying rotten fruits and vegetables
Maximum Marks	4 Marks

Technical Architecture:

A camera-equipped conveyor belt captures images of fruits/vegetables. These images are sent to an AI module using a **VGG16**-based model (transfer learning). The model identifies rotten produce. Based on predictions, sorting actuators separate fresh from rotten produce. The application is hosted locally or optionally on cloud (e.g., Google Colab or AWS).

EXAMPLE:Smart Sorting:Transfer Learning for Identifying rotten fruits and vegetables

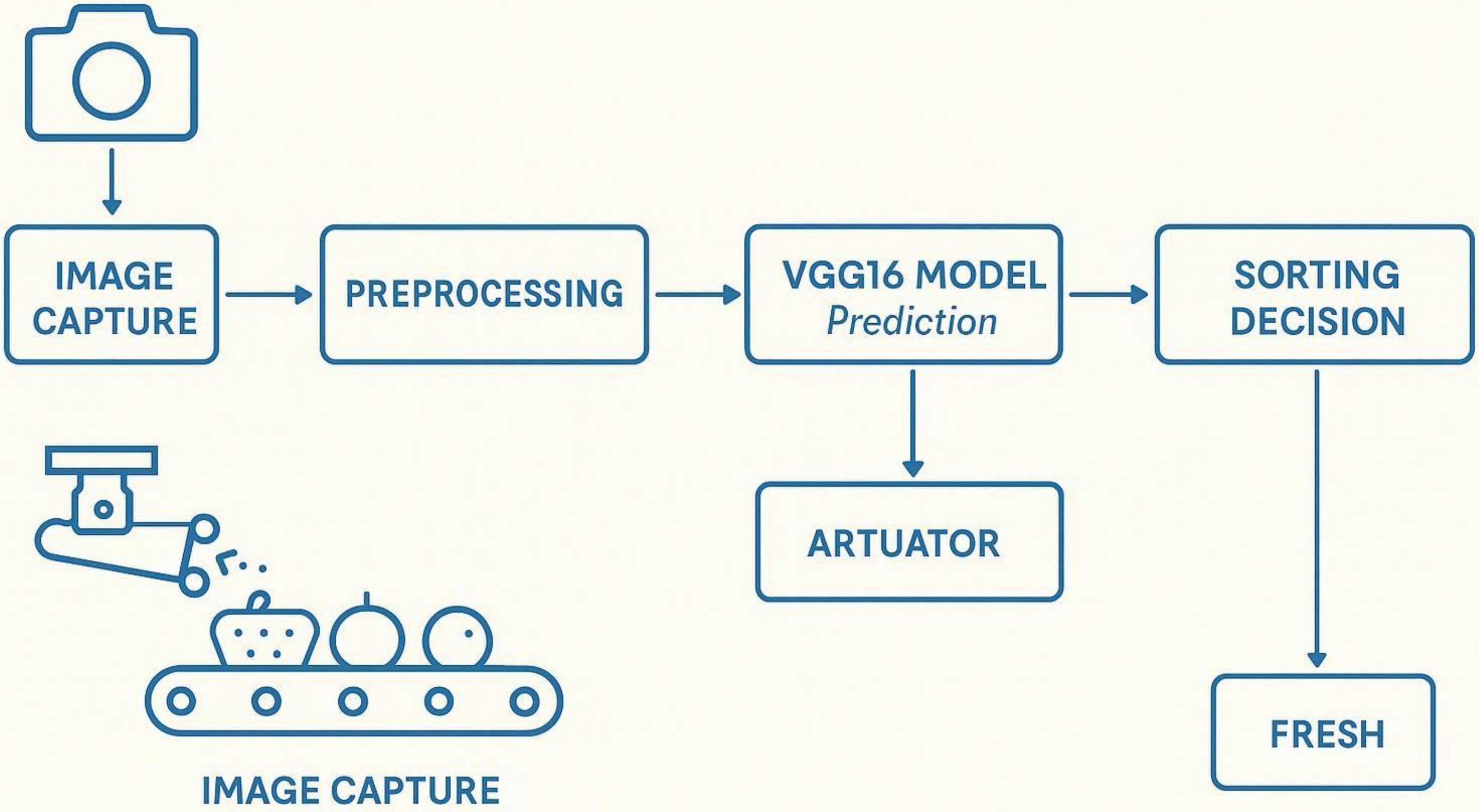
IMAGE CAPTURE
PREPROCESSING
VGG16 MODEL
Prediction
SORTING DECISION
ARTUATOR
FRESH
IMAGE CAPTURE

Table1:Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Interface to upload/test images, optional dashboard	HTML, CSS, Flask, Jupyter Notebook(or)Google colab
2.	Application Logic-1	Image acquisition from camera/conveyor input	OpenCV, Python
3.	Application Logic-2	Preprocessing and augmentation of images	Transfer learning model training & prediction.
4.	Application Logic-3	Transfer learning model training & prediction	Transfer learning model training & prediction
5.	Database	Local dataset of fruit/vegetable images	Directory-based storage; Image folders
6.	Cloud Database	Optional dataset backup/storage	Google Drive, AWS S3 (optional)
7.	File Storage	Stores input/output images and model files locally	Local filesystem / Google Drive (Colab)
8.	External API-1	Optional integration with IoT sensors	REST API (optional, if sensors included)
9.	External API-2	Optional cloud ML API (if used for performance)	Google Cloud Vision API (optional)
10.	Machine Learning Model	Detect rotten fruits using image classification	Transfer Learning using VGG16 (Keras, TensorFlow)
11.	Infrastructure (Server / Cloud)	Transfer Learning using VGG16 (Keras, TensorFlow)	Google Colab / Local Machine / AWS EC2

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks for ML and image processing	Frameworks for ML and image processing
2.	Security Implementations	Not sensitive; limited access locally or via secured cloud	Optional IAM for cloud (e.g., Google IAM)
3.	Scalable Architecture	Modular design; can be expanded with REST APIs & cloud	Flask API (if deployed), AWS, Kubernetes
4.	Availability	High if hosted on cloud; manual mode otherwise	Google Colab / AWS / On-prem systems
5.	Performance	Optimized using pretrained VGG16, image augmentation	VGG16, TensorFlow, Keras