DATE	20-02-2026
TEAM ID	LTVIP2026TMIDS90622
PROJECT NAME	SMART SORTING : TRANSFER LEARNING FOR IDENTIFYING ROTTEN FRUITS AND VEGETABLES
MAXIMUM MARKS	5 MARKS

4.3 SOLUTION ARCHITECTURE

The solution architecture is designed as a modular and scalable deep learning system that integrates image acquisition, preprocessing, transfer learning-based classification, and automated sorting or result display. The architecture ensures real-time detection, high accuracy, and easy deployment in both retail and industrial environments.

1. Image Acquisition Layer

This is the input layer of the system where fruit and vegetable images are captured using cameras, mobile devices, or conveyor belt-mounted vision systems. In real-time industrial setups, a high-resolution camera continuously captures images as items move on a conveyor. In mobile or web-based applications, users can upload images directly through the interface. The captured images are forwarded to the preprocessing module.

2. Data Preprocessing Layer

The preprocessing layer prepares raw images for model input. Images are resized to match the required input dimensions of the pre-trained model (for example, 224×224 pixels). Normalization is applied to scale pixel values. Noise removal and contrast enhancement techniques improve image clarity. Data augmentation such as rotation, flipping, zooming, and brightness variation helps improve model generalization. This step ensures consistent input quality for accurate classification.

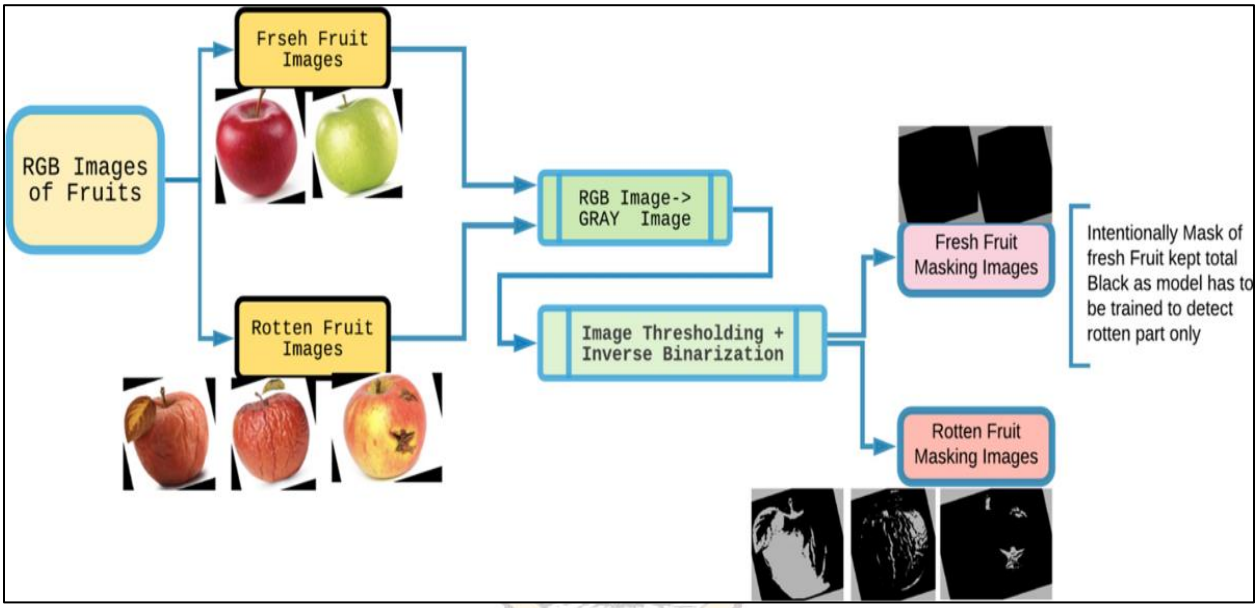
3. Feature Extraction and Transfer Learning Layer

This is the core intelligence layer of the system. A pre-trained Convolutional Neural Network such as MobileNetV2, ResNet50, or VGG16 is used as the base model. These models are originally trained on large datasets like ImageNet and can detect complex visual patterns.

4. Classification Layer

The customized output layer classifies images into predefined categories such as Fresh Apple, Rotten Apple, Fresh Tomato, Rotten Tomato, etc. A Softmax activation function is

used to produce probability scores for each class. The class with the highest probability is selected as the final prediction. Confidence scores can also be displayed for reliability analysis.



5. Decision and Control Layer

Based on the classification result, the system triggers appropriate actions. In a software-based implementation, the result is displayed on a web or mobile dashboard. In an industrial implementation, the output is connected to a microcontroller system that controls actuators or robotic arms to separate rotten items from fresh ones automatically.

6. Deployment Layer

The trained model is deployed using frameworks such as TensorFlow or PyTorch. The application can be hosted on cloud platforms for scalability or deployed on edge devices for faster real-time processing. Edge deployment ensures low latency and reduced dependency on internet connectivity.

Overall Architecture Flow

Image Capture → Image Preprocessing → Feature Extraction (Pre-trained CNN) → Fine-Tuned Classification → Prediction Output → Automated Sorting or Display