Project Design Phase
Solution Architecture

Date	25 June 2025
Team ID	LTVIP2025TMID36697
Project Name	Smart Sorting: Identifying rotten fruits and vegetables using transfer learning
Maximum Marks	4 Marks

Solution Architecture: The Smart Sort Application mainly consists of:

Core Components

1. **Frontend (Presentation Layer)**
    - Built with HTML, CSS, JavaScript
    - Allows users to upload or drag-and-drop images
    - Displays prediction results (label + confidence)
2. **Backend (Application Logic)**
    - Developed using Flask (Python)
    - Handles image reception and routing
    - Connects to the trained deep learning model for inference
3. **Model Layer (AI Engine)**
    - Uses **MobileNetV2** (transfer learning from ImageNet)
    - Preprocessing: resize (224×224), normalize, preprocess input
    - Output: 28-class softmax prediction with high accuracy
4. **Data Layer**
    - Input dataset from Kaggle (28 classes: fresh & rotten produce)
    - Images are augmented and standardized before training
    - .h5 model file packaged with the application
5. **Deployment Layer**
    - App bundled as .exe using PyInstaller for offline use
    - Runs on standalone systems without needing Python
    - Folder structure designed for portable execution

🔁 End-to-End Flow

User Upload ➜ Flask API ➜ Preprocess ➜ Model Predict ➜ Result Return ➜ UI Render

Solution Architecture Diagram:

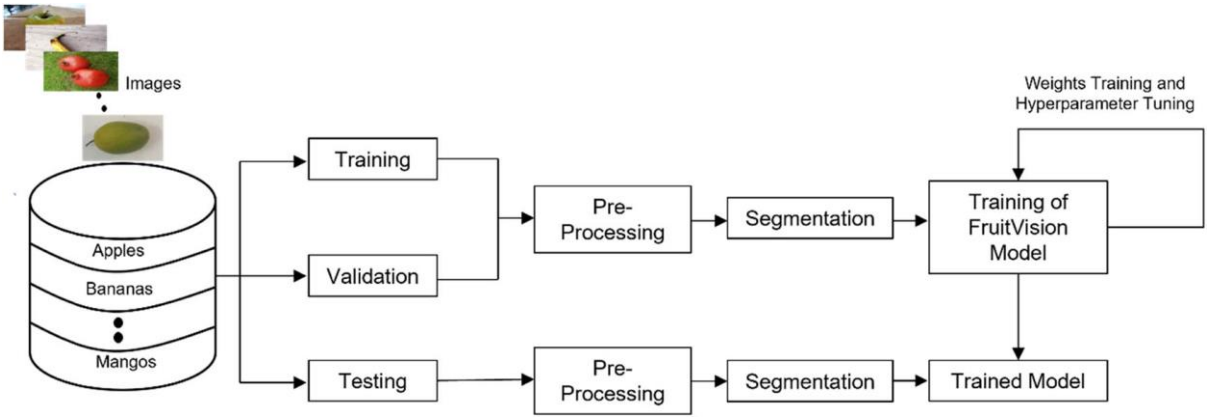

Figure 1: Architecture and data flow of the Model

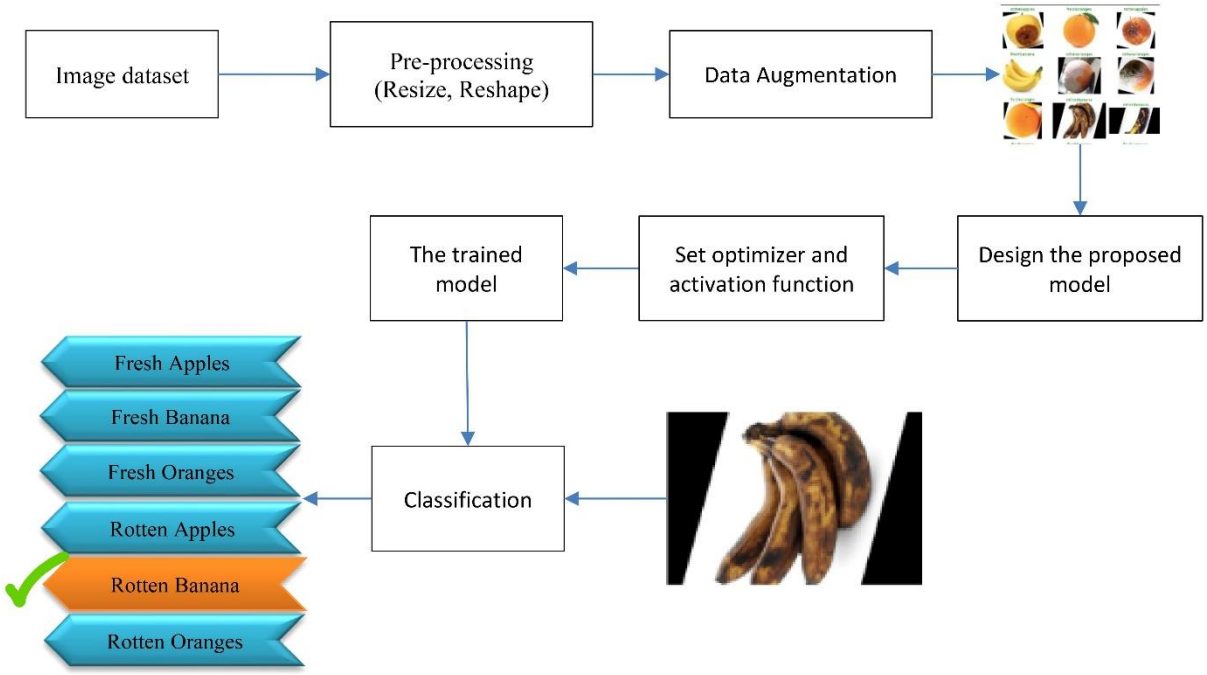

Figure 2: Architecture and data flow of the Smart Sort Application