**Solution Architechture**

| **Field** | **Details** |
| --- | --- |
| Date | 16-06-2025 |
| Team ID | LTVIP2025TMID35102 |
| Project Name | Smart Sorting: Detecting Rotten Fruits with Transfer Learning |
| Maximum Marks | 4 Marks |

**What is Solution Architecture?**

Solution Architecture serves as a blueprint that connects the real-world problem of inefficient manual fruit sorting with an intelligent, automated machine learning system. It bridges challenges in agricultural quality control—such as inconsistent sorting, time-consuming inspections, and post-harvest losses—with technologies like image classification, deep learning, and web applications.

The architecture outlines:  
• The structure of the system including model training, inference pipeline, and web interface  
• How fruit images are processed and classified from input to result  
• Technologies and frameworks used (e.g., TensorFlow/Keras, Flask)• Deployment and usage strategy for end-users like farmers or vendors

**Goals of the Solution Architecture for This Project:**

• Enable accurate classification of fresh vs. rotten fruits using image recognition  
• Provide a simple web interface for real-time fruit image uploads and predictions  
• Ensure scalability for different types of fruits and deployment environments  
• Minimize model training requirements using transfer learning  
• Allow future integration with mobile apps or IoT-based edge devices for real-time field use

**Core Components of the Architecture**

| Layer | Component | Description |
| --- | --- | --- |
| Data Acquisition | Image Upload (via Web UI) | Users upload fruit images through a web interface built using Flask. |
| Preprocessing | Image Resizing & Normalization | Uploaded images are resized and normalized to match model input specifications. |
| Model Inference | CNN Model (VGG16-based) | A pretrained deep learning model classifies images as fresh or rotten. |
| Backend Logic | Flask API Server | Handles file uploads, passes data to the model, and returns predictions. |
| Frontend Interface | HTML Templates + Bootstrap | Simple, user-friendly interface for uploading images and viewing results. |
| Security & Access | Local Authentication (optional) | Access can be restricted to specific users if needed. |
| Deployment | Localhost / Cloud / Docker | Can be deployed locally or to cloud environments (AWS, Heroku, etc.). |

**Data Flow within the System**

**User Journey and Data Movement:**

1. Image Upload: Users upload an image of a fruit through the web application.
2. Preprocessing: The image is resized and normalized to suit the model's input layer.
3. Prediction: The CNN model (VGG16-based) processes the image and classifies it as "Fresh" or "Rotten."
4. Result Display: The prediction result is returned and displayed on the web interface.
5. (Optional) Logging and Storage: Data may be logged for future improvements or audits.

**Summary**

**This solution architecture ensures:**  
• Seamless integration of machine learning with a web interface for real-time use  
• Accurate and fast detection of fruit quality to reduce food waste  
• Scalability to multiple fruit types, languages, and user roles  
• Flexibility to run on both local systems and cloud platforms  
• A foundation for future development like mobile app integration or IoT-based automation

By combining transfer learning with intuitive web deployment, the architecture delivers a smart and accessible tool to modernize fruit sorting and support farmers, vendors, and consumers with better quality control.