

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

Date	25 th June 2025
Team ID	LTVIP2025TMID34982
Project Name	Smart Sorting using transfer learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline machine

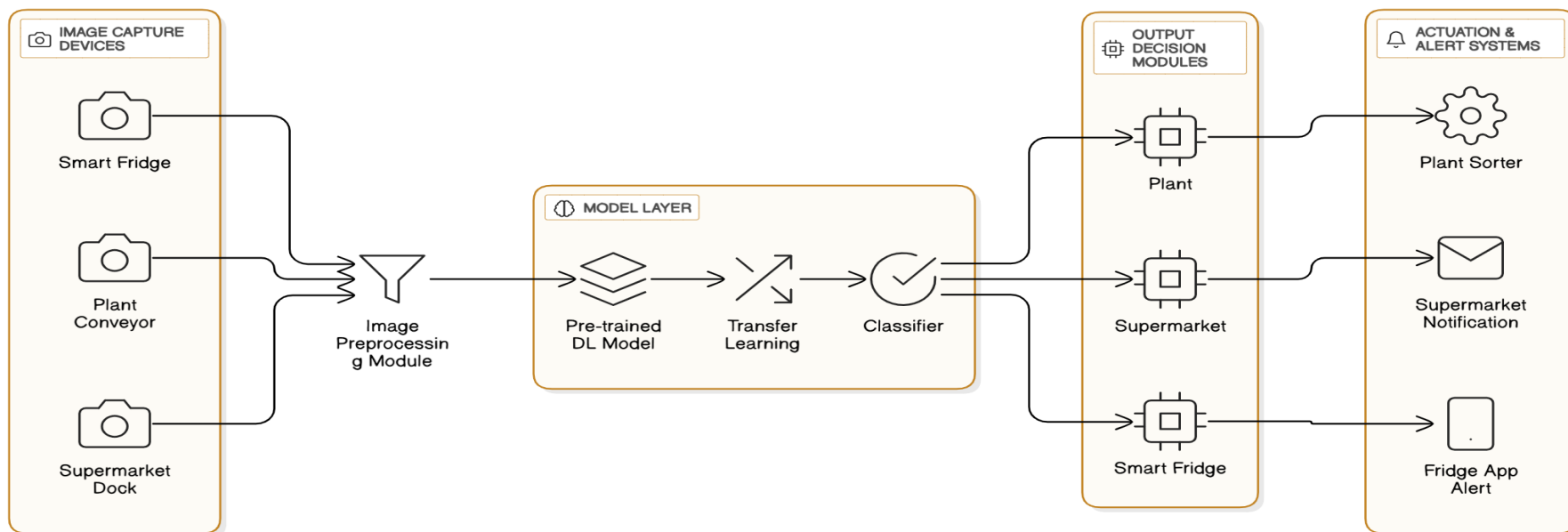


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI for uploading images and viewing predictions	HTML, CSS, JavaScript, Bootstrap
2.	Application Logic-1	Flask-based server logic to handle upload, model prediction	Python, Flask
3.	Application Logic-2	Image preprocessing and resizing before feeding to model	OpenCV, TensorFlow/Keras
4.	Application Logic-3	Not used in this project	-
5.	Database	No traditional DB required; optional logging via flat files	JSON / CSV (optional), SQLite (if expanded)
6.	Cloud Database	Not applicable (runs locally or on personal system)	-
7.	File Storage	Uploaded images and trained .h5 model file	Local Filesystem
8.	External API-1	Not used in this project	-
9.	External API-2	Not used in this project	-
10.	Machine Learning Model	To classify fruits/vegetables as healthy or rotten using image data	VGG16 (Transfer Learning) – Keras / TensorFlow
11.	Infrastructure (Server / Cloud)	Application runs on user system using Anaconda or can be deployed to cloud	Local (Flask), Cloud-compatible (Heroku, Render)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Uses open libraries and frameworks	Flask (Python), Keras, TensorFlow, Bootstrap
2.	Security Implementations	Restricts file type upload, basic validation, no external sharing of files	File type validation, Flask secure upload methods
3.	Scalable Architecture	Can be scaled by deploying on cloud with retraining options	Flask Microservice; Cloud container ready
4.	Availability	Local app, can be made highly available via cloud and CDN	Render / Heroku / AWS or Azure Hosting (optional)
5.	Performance	Fast image prediction with pre-loaded model; small load handled easily	TensorFlow optimized, low latency, local model use