**Data Flow Diagram**

| **Field** | **Details** |
| --- | --- |
| Date | 16-06-2025 |
| Team ID | LTVIP2025TMID35102 |
| Project Name | Smart Sorting: Detecting Rotten Fruits with Transfer Learning |
| Maximum Marks | 2 Marks |

This Data Flow Diagram illustrates how data moves through the Smart Sorting system — from user interaction (image submission) to prediction and result display — using a Transfer Learning model for real-time fruit classification. It maps out the internal components such as preprocessing modules, data storage units, and ML models, showing how each layer communicates to deliver intelligent, accurate, and user-friendly results.

**Level 0 – Context-Level DFD**

At the highest level, the system involves the following components:

| **Entity / Process** | **Description** |
| --- | --- |
| User (Farmer/Vendor) | Uploads fruit image via UI (e.g., drag & drop or file upload) to check whether a fruit is fresh or rotten. |
| Smart Sorting System | Accepts image input, processes it through a deep learning model, and returns the freshness status. |
| ML Developer/Admin | Manages dataset updates, model retraining, and performance optimization (not directly visible to end user). |

The User sends an image to the Smart Sorting System, and the system returns a classification result.

**Level 1 – Detailed DFD Explanation**

This level breaks down the internal components and interactions within the system:

| **Steps** | **Component** | **Functionality** |
| --- | --- | --- |
| 1 | **Image Input Module** | The user uploads or captures an image of a fruit via a simple web UI. The module checks file validity (format, size). |
| 2 | **Preprocessing Module** | Resizes the image (e.g., to 224x224), normalizes pixel values, and converts it into an array suitable for the CNN model. |
| 3 | **ML Classification Engine** | Uses a pre-trained CNN model (like VGG16 with transfer learning) to predict whether the fruit is Fresh or Rotten, along with a confidence score. |
| 4 | **Prediction Output Handler** | Displays the result to the user in a clean format — e.g., a label with color code (Green for Fresh, Red for Rotten) and probability percentage. |
| 5 | **(Optional) Log Storage** | Saves image and prediction details for future analysis or dataset expansion (used only if logging is enabled for feedback/retraining). |
| 6 | **Admin Panel (Backend)** | Allows model updates, dataset additions, or retraining operations. This layer ensures the system can evolve and improve over time. |

**Data Stores**

| Data Store | Purpose |
| --- | --- |
| Fruit Image Dataset | Contains labeled images used to train/test the transfer learning model. |
| Prediction Log (Optional) | Stores user image submissions and prediction results for future refinement. |

**Data Flow Summary**

**Input Flow:**

User submits a fruit image → Image is preprocessed → Model performs classification → Output is displayed to the user.

**Output Flow:**

The classification result (e.g., "Rotten" – 94%) is shown instantly for real-time sorting or rejection.

**Optional Admin Flow:**

Admin can access stored logs, upload new training data, and retrain the model to improve accuracy.