Project

Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables

Team ID: LTVIP2025TMID45424

Team Members:

- Tellapuri Hiranya Jahnavi Latha (Team Lead)
- B Haritha
- Sidda Keerthi
- Chantimalla Bharathi

Phase 1: Brainstorming and Ideation

Objective: To conceptualize a practical and efficient solution for rotten fruits and vegetables classification using deep learning methodologies, with a focus on medical relevance and real-time deployment potential.

Key Points:

- **Problem Statement:** Manual identification of rotten fruits and vegetables is time-consuming, prone to errors, and requires expert pathologists.
- **Proposed Solution:** Develop a deep learning model using transfer learning to accurately classify rotten fruits and vegetables.
- **Target Users:** This type of classification is used in food factories which uses fruits and vegetables.
- **Expected Outcome:** A web-based application capable of accurately classifying rotten fruits and vegetables using an uploaded image, improving accuracy and efficiency.

Phase 2: Requirement Analysis

Objective: To identify all technical and functional needs essential for the development and deployment of the rotten fruits and vegetables classification system.

Key Points:

- **Technical Requirements:**

  ○ Python 3.9 or 3.10 ○ TensorFlow >= 2.10 ○

  Flask (for web app integration) ○ NumPy, Pandas, Matplotlib, OpenCV, Seaborn ○ Anaconda

/ Google Colab / VS Code (for development)

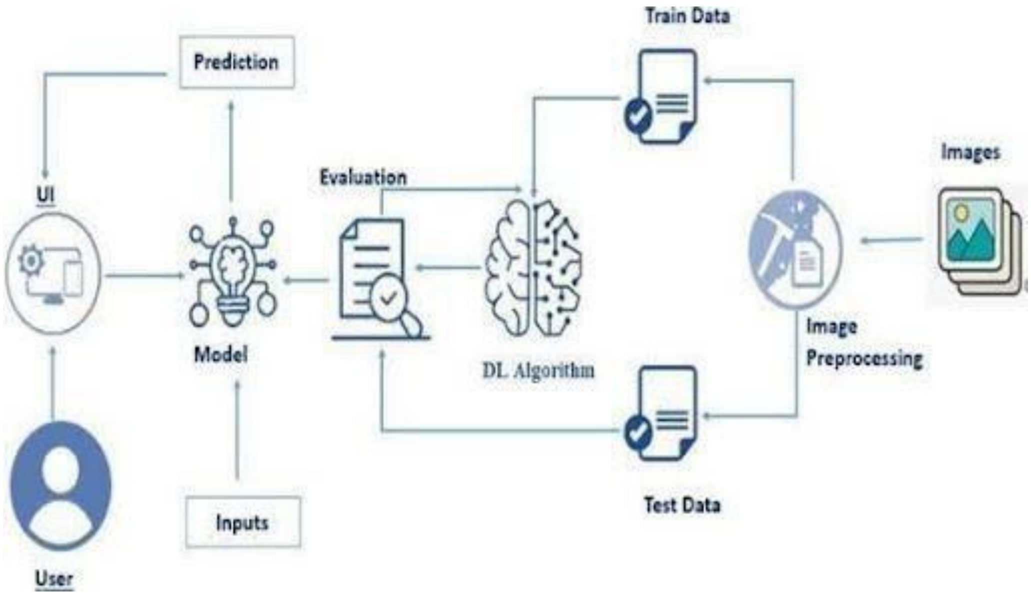
- **Functional Requirements:**
  - User uploads a blood fruits ahd vegetables image
  - The system preprocesses the image
  - The trained model predicts whether the fruit or vegetable is ripened or not
  - Result is displayed with classification confidence
- **Constraints & Challenges:**
  - Model performance depends on dataset quality
  - File compatibility for .h5 in different environments
  - Limitations on Colab runtime and file storage
  - Hardware limitations on local machines for training

Phase 3: Project Design

Objective: To establish a clear and scalable system architecture and define user flow for seamless application interaction.

Key Points:

- **System Architecture:**



- **User Flow:**

- User accesses the application (locally or via browser)
- Uploads image of blood cell
- Backend model processes and classifies the image
- Result is shown on the frontend

Phase 4: Project Planning

Objective: To outline the project timeline, task distribution, and dependencies.

Sprint	***Task***	***Priority***	***Duration***	***Deadline***	***Assigned to***	***Dependencies***	***Expected outcome***
Sprint 1	*Environment Setup & Package Installation*	*High*	*3 hours*	*Day 1*	*Member 1*	*Anaconda, Python*	*Project environment ready*
Sprint 1	*Dataset Collection & Preprocessing*	*High*	*4 hours*	*Day 1*	*Member 2*	*Dataset access*	*Clean, prepared image dataset*
Sprint 2	*Model Building using Transfer Learning*	*High*	*5 hours*	*Day 2*	*Member 3*	*Preprocessed data, TensorFlow*	*Trained classification model*
Sprint 2	*Flask Web App Integration*	*Medium*	*3 hours*	*Day 2*	*Member 1 & 4*	*Trained Model, Flask installed*	*Working web interface*
Sprint 3	*Testing & Debugging*	*Medium*	*2 hours*	*Day 2*	*Member 2 & 3*	*Complete System*	*Bug-free and responsive system*
Sprint 3	*Final Presentation & Deployment*	*Low*	*1 hour*	*End of Day 2*	*Entire Team*	*Working application*	*Project deployed and demo-ready*

Phase 5: Project Development

Objective: To build and test the system iteratively, addressing challenges and refining model performance.

Key Points:

- **Technology Stack Used:**
  - Language: Python
  - Frameworks: TensorFlow, Flask
  - Tools: Google Colab, Anaconda, Jupyter Notebook, VS Code
- **Development Process:**
  1. Data preprocessing and augmentation
  2. Building and training model using transfer learning (e.g., MobileNetV2, ResNet50)
  3. Model evaluation and tuning
  4. Web application integration using Flask
  5. Deployment and testing
- **Challenges and Fixes:**
  - Issue: .h5 file not opening in VS Code
    - ✦ Fix: Used correct Python environment with TensorFlow installed (Python 3.9/3.10)
  - Issue: Jupyter notebook not launching from virtual environment
    - ✦ Fix: Installed Jupyter inside the specific conda environment and added kernel
      - Issue: TensorFlow install errors
    - ✦ Fix: Switched to supported Python version (3.10) and used clean virtual environment