

Project

Enchanted Wings: Marvels of Butterfly Species

Team ID: LTVIP2025TMID44932

Team Members:

- Batta Sai Vardhan (Team Lead)
- Palle Mohan Krishna
- H Kiran Kumar
- A Charan Venkat

Phase 1: Brainstorming and Ideation

Objective: To conceptualize a practical and efficient solution for butterfly image classification using transfer learning methodologies, with a focus on medical relevance and real-time deployment potential.

Key Points:

- **Problem Statement:** Manual identification of butterfly image 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 forest officers which is to used to detect the species of the butterfly.
- **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 butterfly image classification system.

Key Points:

- **Technical Requirements:**
 - Python 3.9 or 3.10

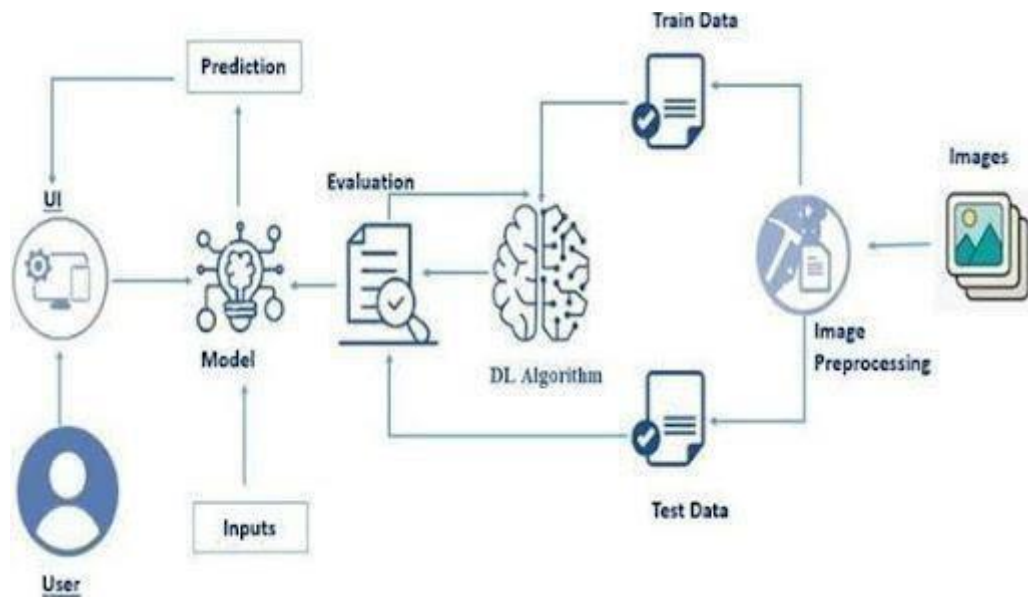
- TensorFlow >= 2.10
- Flask (for web app integration)
- NumPy, Pandas, Matplotlib, OpenCV, Seaborn
- Anaconda / Google Colab / VS Code (for development)
- **Functional Requirements:**
 - User uploads a blood fruits and vegetables image.
 - The system preprocesses the image.
 - The trained model predicts the buter fly belongs to which species.
 - 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.

<i>Sprint</i>	<i>Task</i>	<i>Priority</i>	<i>Duration</i>	<i>Deadline</i>	<i>Assigned to</i>	<i>Dependencies</i>	<i>Expected outcome</i>
<i>Sprint 1</i>	<i>Environment Setup & Package Installation</i>	<i>High</i>	<i>3 hours</i>	<i>Day 1</i>	<i>Member 1</i>	<i>Anaconda, Python</i>	<i>Project environment ready</i>
<i>Sprint 1</i>	<i>Dataset Collection & Preprocessing</i>	<i>High</i>	<i>4 hours</i>	<i>Day 1</i>	<i>Member 2</i>	<i>Dataset access</i>	<i>Clean, prepared image dataset</i>

<i>Sprint 2</i>	<i>Model Building using Transfer Learning</i>	<i>High</i>	<i>5 hours</i>	<i>Day 2</i>	<i>Member 3</i>	<i>Preprocessed data, TensorFlow</i>	<i>Trained classification model</i>
<i>Sprint 2</i>	<i>Flask Web App Integration</i>	<i>Medium</i>	<i>3 hours</i>	<i>Day 2</i>	<i>Member 1 & 4</i>	<i>Trained Model, Flask installed</i>	<i>Working web interface</i>
<i>Sprint 3</i>	<i>Testing & Debugging</i>	<i>Medium</i>	<i>2 hours</i>	<i>Day 2</i>	<i>Member 2 & 3</i>	<i>Complete System</i>	<i>Bug-free and responsive system</i>
<i>Sprint 3</i>	<i>Final Presentation & Deployment</i>	<i>Low</i>	<i>1 hour</i>	<i>End of Day 2</i>	<i>Entire Team</i>	<i>Working application</i>	<i>Project deployed and demo-ready</i>

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