project

Enchanted Wings: Marvels of Butterfly Species Classification Using Deep Learning

Team ID: LTVIP2025TMID45420
Team Members:

Team Leader: Shaik Mahammad Shahid Afrid

Team Member: Meda Dhruva Teja

Team Member: B Hruday

Team Member: C Ajay Kumar

Phase 1: Brainstorming and Ideation

Objective:
To build an automated and intelligent butterfly species classification system using deep learning that promotes biodiversity awareness and aids entomologists in species identification.

Key Points:

- Problem Statement: Manual identification of butterfly species is error-prone, time-consuming, and requires expertise.
- Proposed Solution: Use a convolutional neural network (CNN) to automatically classify butterfly images into their respective species.
- Target Users: Biodiversity researchers, educational institutions, wildlife photographers, and conservationists.
- Expected Outcome: A web application that classifies butterfly images with high accuracy and displays species names with confidence scores.

Phase 2: Requirement Analysis

- Objective:

  To identify all technical and functional requirements for building and deploying the butterfly classifier.

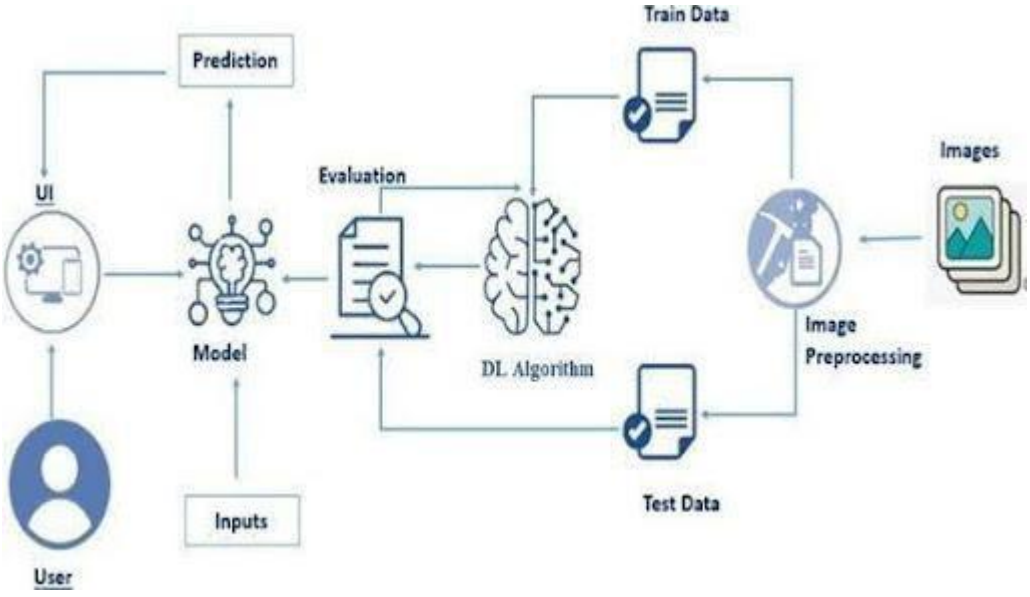
- 🧪 Technical Requirements:
- Python 3.9+
- TensorFlow 2.12.0+

- Flask
- NumPy, Pandas, OpenCV, Matplotlib, Pillow
- Google Colab / Jupyter Notebook / VS Code
- 🔧 Functional Requirements:
- Upload butterfly image
- Preprocess and feed image to model
- Predict species name using trained .h5 model
- Display result with confidence score
- ⚠️ Constraints & Challenges:
- Quality and balance of dataset affects performance
- Colab memory and runtime limits
- .h5 model compatibility between environments
- Training time on limited hardware

Phase 3: Project Design

Objective: To design a clear architecture and seamless user experience for butterfly classification.Key Points:

- System Architecture:



- User Flow:

  User opens the web app

- Uploads a butterfly image
- Flask backend loads .h5 model and classifies the image
- Result is shown with class name and confidence

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 iteratively develop the model and web interface, ensuring accuracy and stability.

Key Points:

- Technology Stack Used:

- Language: Python
- Libraries: TensorFlow, Keras, NumPy, Pillow, OpenCV
- Frontend: HTML, CSS
- Backend: Flask
- Tools: Google Colab (training), VS Code (deployment)

Development Process:

1. Dataset loaded from Kaggle and extracted
2. Data preprocessing and augmentation in Colab
3. CNN Model built using Conv2D layers
4. Trained and evaluated with .h5 output
5. Flask web app developed with file upload and prediction
6. Deployed locally with working UI

Challenges and Fixes:

- Issue: .h5 model not loading in Flask
  Fix: Matched TensorFlow version (2.12.0) in virtual environment
- Issue: Virtual environment errors in VS Code
  Fix: Activated venv39 manually and installed required packages
- Issue: Model trained on Colab was slow
  Fix: Optimized batch size and used early stopping
- Issue: Dataset contained improperly labeled images
  Fix: Filtered test images and manually verified directory structure

Final Outcome:

- Working CNN model trained to classify butterfly species
- Flask web app for uploading and predicting butterfly images
- Project packaged and ready for submission with .h5 model
- Documentation and sprint planning completed

1. Model evaluation and tuning
2. Web application integration using Flask
3. 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

Final Output:

Images: