

1. Introduction

Project Title: Enchanted Wings: Marvels of Butterfly Species

Team Members:

1. Venkanna Babu
2. Vardhineedi Dharun Sai Ranjith
3. Vakalapudi Halekya Devi
4. Tsavatapalli Venkata Sivasankar Varaprasad

2. Project Overview

Purpose:

This project aims to automate the identification and tracking of butterfly species using deep learning models

Features:

- Automated butterfly species classification
- Real-time image processing and predictions
- Flask-based web interface for user interaction
- Educational and citizen science support

3. Architecture

The project follows a deep learning-based pipeline:

- User uploads butterfly image through UI.
- Image Preprocessing: Image is processed and normalized.
- DL Algorithm: Pre-trained VGG16 model is used for feature extraction and classification.
- Evaluation: Model evaluates the image and predicts the butterfly species.
- Result Display: Result is displayed to the user.

4. Setup Instructions

Prerequisites:

- Python 3.x
- Flask
- TensorFlow/Keras
- OpenCV
- Pre-trained VGG16 model

Installation:

1. Clone the repository.
2. Install dependencies: `pip install -r requirements.txt`. Place the `vgg16_model.h5` file in the project directory.
4. Run the Flask server: `python app.py`

5. Folder Structure

PROJECTS/

```
■■■ client/ (React Frontend)
■   ■■■ public/
■   ■■■ src/
■   ■   ■■■ components/
■   ■   ■■■ pages/
■   ■   ■■■ App.js
■   ■   ■■■ index.js
■   ■■■ package.json
■■■ server/ (Flask Backend)
■   ■■■ static/
■   ■■■ templates/
■   ■■■ app.py
■   ■■■ vgg16.model.h5
■   ■■■ requirements.txt
■■■ README.md
```

Explanation:

- client: React frontend containing components and pages.
- server: Flask backend with model and API logic.
- README.md: Project documentation and setup instructions.

6. Running the Application

Frontend:

```
cd client
npm start
```

Backend: cd server python
app.py

7. API Documentation

Endpoints:

POST /api/identify - Upload butterfly image for identification

Request: Form-data with image file

Response: JSON with species information and confidence score

GET /api/species - Retrieve all species information

Response: JSON array of species data

8. Authentication

- JWT (JSON Web Tokens) for user authentication
- Protected routes for user-specific features
- Password hashing with bcrypt

9. User Interface

Screenshots:

- Landing page with upload functionality
- Identification results page
- Species information page
- User dashboard

10. Testing

Testing Strategy:

- Manual testing of the image upload and prediction process.- Accuracy validation using a test dataset.

Tools Used:

- Flask (web routing)
- Python (manual testing)
- Jest/React Testing Library for frontend components

11. Screenshots or Demo

Demo Link: https://drive.google.com/file/d/1_jgQuG7QjAYPg1okmiSAiLq2IsBjL9F/view?usp=sharing

12. Known Issues

- Limited species database.
- Accuracy may decrease with poor quality or noisy images.
- Model may misclassify species with similar visual patterns.

13. Future Enhancements

- Expand dataset to cover more butterfly species.
- Implement user authentication and save search history.
- Develop a mobile application for real-time field identification.
- Improve model accuracy using more advanced architectures.