Enchanted Wings: Marvels of Butterfly Species

Team Members:

TEAM ID: LTVIP2025TMID59656

| **MEMBER** | **RESPONSIBILITY** |
| --- | --- |
| 1. Mudikonda Praneetha | Frontend and algorithms |
| 2. Kola Divya Surya Chandrika | Datasets gathering |
| 3. Kesani Phani Sekhar | Testing & Documentation. |
| 4. Kukkala Naga Tulasi | Testing &Documentation. |

# 1. Introduction

Project Title:

Enchanted Wings-Marvels of Butterfly Species

# 2. Project Overview

***Purpose:***  
The goal of this project is to develop a robust butterfly species classification system using transfer learning techniques on a dataset of butterfly images. This application identifies the species from images using a pre-trained deep learning model.

***Features:***  
- Upload butterfly images and predict species.  
- Use of pre-trained CNNs (like MobileNetV2) for efficient image classification.  
- Supports over 75 butterfly species.  
- Responsive user interface for easy usage.

# 3. Architecture

***Frontend:***  
- Built using HTML, CSS, and JavaScript for simplicity.  
- Can be extended with React for dynamic rendering.

***Backend:***  
- Built with Python using Flask.  
- Serves the model inference API.

***Database:***  
- Not used in the current version (No data persistence).  
- Can be extended with MongoDB for storing user inputs or predictions.

# 4. Setup Instructions

***Prerequisites:***  
- Python 3.x  
- Flask  
- TensorFlow / Keras  
- NumPy, OpenCV, Pillow  
- Jupyter Notebook or Google Colab (for model training)

***Installation:***  
```bash  
git clone

https://github.com/Praneetha612/ENCHANTED-WINGS-MARVELS-OF-BUTTERFLY-SPECIES.git

cd butterfly-classifier  
pip install -r requirements.txt  
```

# 5. Folder Structure

Butterfly-Classifier/  
├── model/ # Saved trained model (.h5 or .pkl)  
├── dataset/ # Dataset of butterfly images  
├── static/ # Static assets for frontend  
├── templates/ # HTML templates for Flask  
├── app.py # Flask backend  
├── train\_model.ipynb # Jupyter Notebook for training  
└── requirements.txt # Python dependencies

# 6. Running the Application

Start the Flask Backend:  
```bash  
python app.py  
```

Open your browser and go to: http://127.0.0.1:5000/

# 7. API Documentation

POST /predict  
Description: Accepts an image file and returns the predicted butterfly species.  
Request: multipart/form-data  
- file: image file  
Response:  
{  
 "prediction": "Papilio demoleus",  
 "confidence": "98.5%"  
}

# 8. Authentication

No authentication is implemented in this version.  
Can be integrated in future using JWT or session-based auth if user login is required.

# 9. User Interface

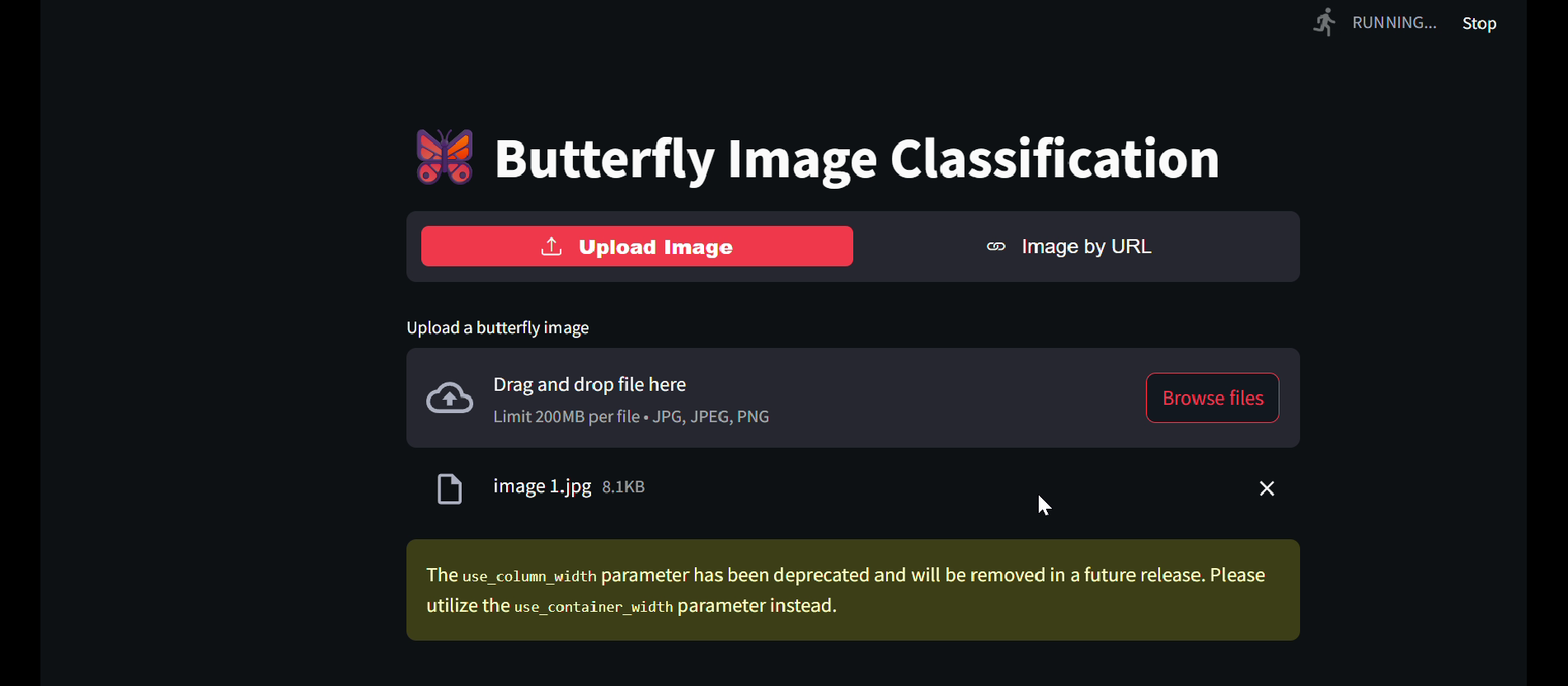
Screenshots:  
- Upload image interface  
- Prediction result display

# 10. Testing

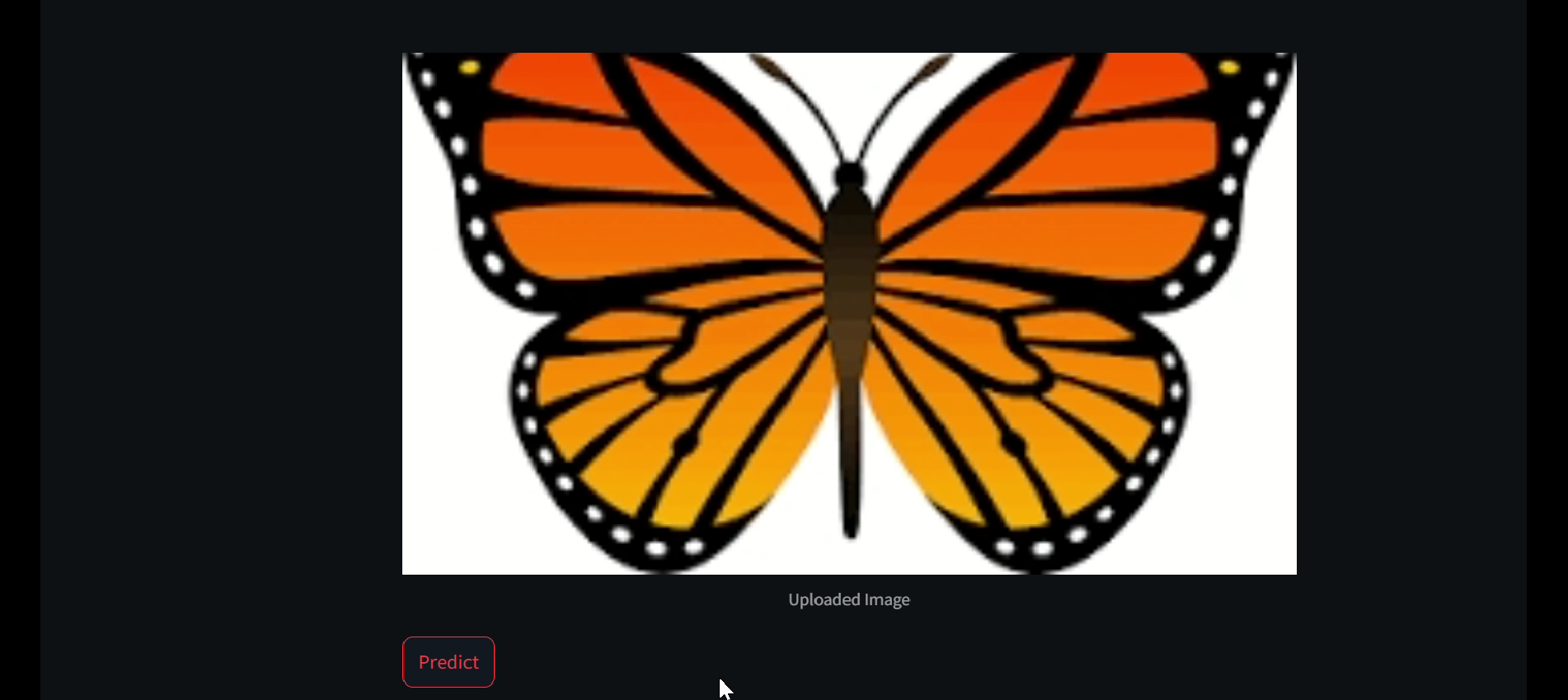
Manual Testing:  
- Tested with random butterfly images from web and dataset.  
- Compared predicted output with known labels.  
Tools:  
- Jupyter Notebook visualizations  
- TensorBoard (for training metrics)

# 11. Screenshots or Demo

Screenshots:  
- Home Page



- Prediction Result



# 12. Known Issues

- Accuracy may degrade for unseen or poor-quality images.  
- Slow performance for large image files on low-end systems.

# 13. Future Enhancements

- Extend UI using React.  
- Add login/signup features.  
- Add image augmentation to improve model robustness