

Artificial Intelligence and Machine Learning

Project Documentation

1. Introduction

- **Project Title:** Enchanted Wings: Marvels of Butterfly Species

- **Team Members:**

Team Leader : Chintapatla Varsha

Team member :Prasadam Supriya

Team member :Kaduru Pavithra

2. Project Overview

- **Purpose:**

The goal of this project is to develop an intelligent web-based butterfly classification tool that identifies butterfly species from uploaded images using deep learning (CNN/VGG16). It aims to assist researchers, educators, and enthusiasts in accurately identifying species, promoting biodiversity awareness, and supporting conservation efforts through an accessible AI-powered interface.

- **Features:**

1.  **Image Upload Interface**

– Simple and responsive form to upload butterfly images for classification.

2.  **Species Prediction with Pre-trained CNN Model**

– Uses a trained deep learning model (e.g., VGG16) to identify butterfly species from images.

3.  **Prediction Output with Name & Insights**

– Displays predicted butterfly name with aesthetic design and a relevant quote or fun fact.

4.  **Responsive Multi-Page Web Interface**

– User-friendly, mobile-compatible site with navigation between home, input, and result pages.

5.  **Educational Content on Butterfly Diversity**

– Includes butterfly facts, species list, and conservation messages to spread ecological awareness.

3. Architecture – Enhanced Wings: The Marvels of Wings

Frontend:

Developed using HTML and CSS, with optional Bootstrap for styling.

The interface includes multiple pages such as:

- Home – introducing the project and its purpose
- Input – allowing users to upload butterfly images
- Result – displaying the predicted butterfly species
- About – providing educational information and project context

This project does not use frameworks like React; a lightweight, static frontend approach was preferred for simplicity and compatibility.

Backend:

Powered by Flask (Python), the backend handles:

- Image upload and preprocessing
- Serving the trained CNN model (VGG16)
- Predicting butterfly species from the input image
- Routing between web pages

The model is loaded from a .h5 file and used in real-time during user interaction.

Database:

No permanent database is integrated in the current version.

All operations are session-based and processed in-memory during runtime.

In the future, integration with MongoDB or PostgreSQL could be added to:

- Log user predictions
- Track most frequently identified species
- Enable user-specific history or insights

4. Setup Instructions

Prerequisites & Installation – Enhanced Wings

Prerequisites:

- Python 3.9+
- Flask (for backend server)
- Jupyter Notebook / Google Colab (for training and testing the model)
- Required Python Libraries:
 - tensorflow / keras (for deep learning)
 - numpy
 - pillow
 - flask
 - werkzeug

- o gunicorn (*optional for deployment*)
-

⚙️ Installation Instructions:

1. Clone the Repository

bash

CopyEdit

```
git clone https://github.com/your-repo/enhanced-wings
```

```
cd enhanced-wings
```

2. Set Up Virtual Environment (optional but recommended)

bash

CopyEdit

```
python -m venv venv
```

```
source venv/bin/activate      # For Linux/macOS
```

```
venv\Scripts\activate        # For Windows
```

3. Install Required Libraries

bash

CopyEdit

```
pip install -r requirements.txt
```

4. Run the Flask App

bash

CopyEdit

```
python app.py
```

🔗 Open your browser and navigate to: <http://127.0.0.1:5000>

5. Folder Structure

enhanced-wings/

```
|   └── model/
```

```
|       └── butterfly_model.h5      # Trained Keras/TensorFlow model
```

```
|       └── class_labels.json     # Mapping of class indices to butterfly names
```

```
|
```

```
|── static/
|   ├── images/          # Uploaded and background images
|   |   └── butterfly-bg.jpg
|   └── css/
|       └── style.css    # Optional external stylesheet
|
|── templates/
|   ├── index.html      # Homepage (Welcome page)
|   ├── input.html       # File upload / Prediction form
|   └── output.html     # Prediction result page
|
|── app.py              # Flask backend application
|── requirements.txt    # Python dependency file
|── README.md           # Project documentation and usage guide
└── utils.py            # Optional helper functions (e.g., image handling, class mapping)
```

6. Running the Application

1. Activate virtual environment (optional but recommended):

bash

Copy code

```
python -m venv venv
```

```
venv\Scripts\activate # On Windows
```

```
# OR
```

```
source venv/bin/activate # On Mac/Linux
```

2. Install dependencies:

bash

Copy code

```
pip install -r requirements.txt
```

3. Start the Flask server:

bash

Copy code

```
python app.py
```

4. Access your web application:

cpp

Copy code

<http://127.0.0.1:5000/>

7. API Documentation

<http://127.0.0.1:5000/>

8. Authentication

Currently, the Enhanced Wings application does **not implement authentication**, as it is a publicly accessible educational tool.

Future Authentication Scope:

- Role-based access for:
 -  Researchers (to upload new species)
 -  Teachers/Students (to view history of predictions)
- JWT or OAuth2 integration for secure API access
- Admin dashboard to manage butterfly classes and user log

9. User Interface

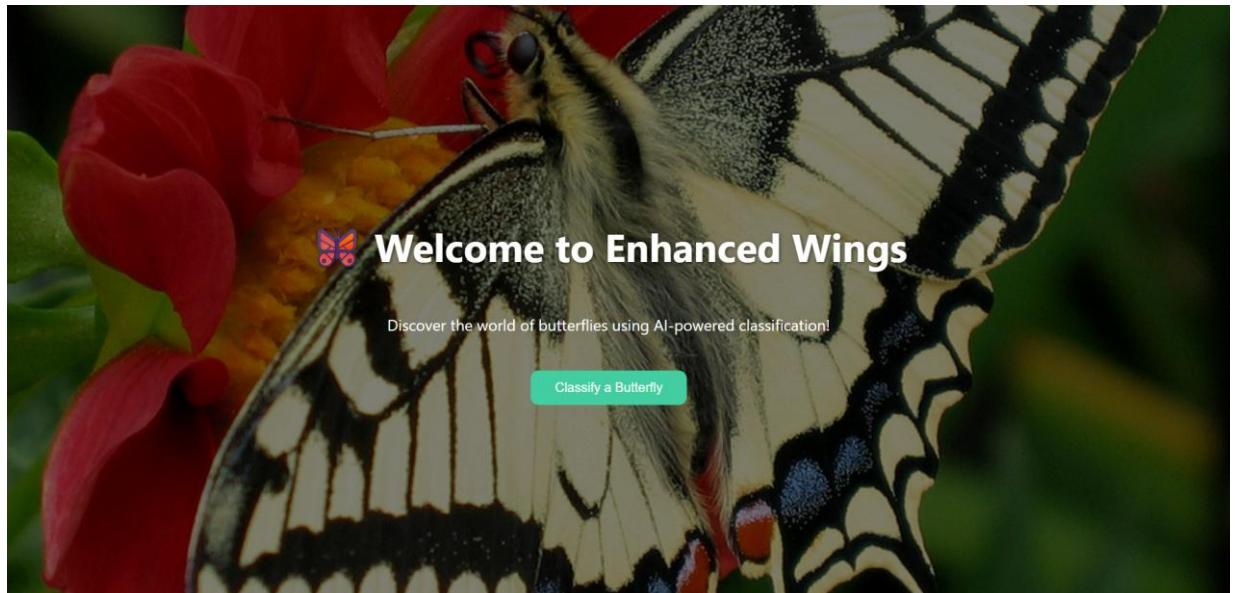
- Clean, multi-page layout (**Home, Prediction, Info, About**)
- Intuitive navigation bar
- Visual badges for classification results
- Input form with validation for image upload

10.Testing:

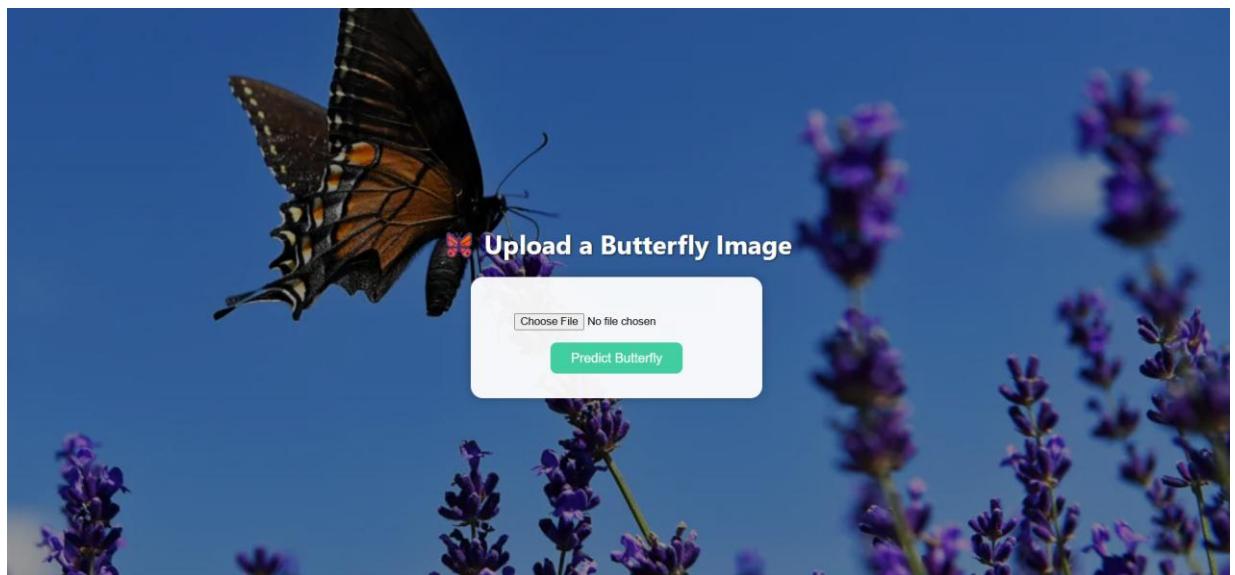
- 1. Model Testing:** Used `train_test_split`, confusion matrix, and classification report (via scikit-learn)
- 2. Tools:** TensorFlow, Keras
- 3. UI Testing:** Manually tested page flow, form behavior, and prediction response

11.screenshots or Demo:

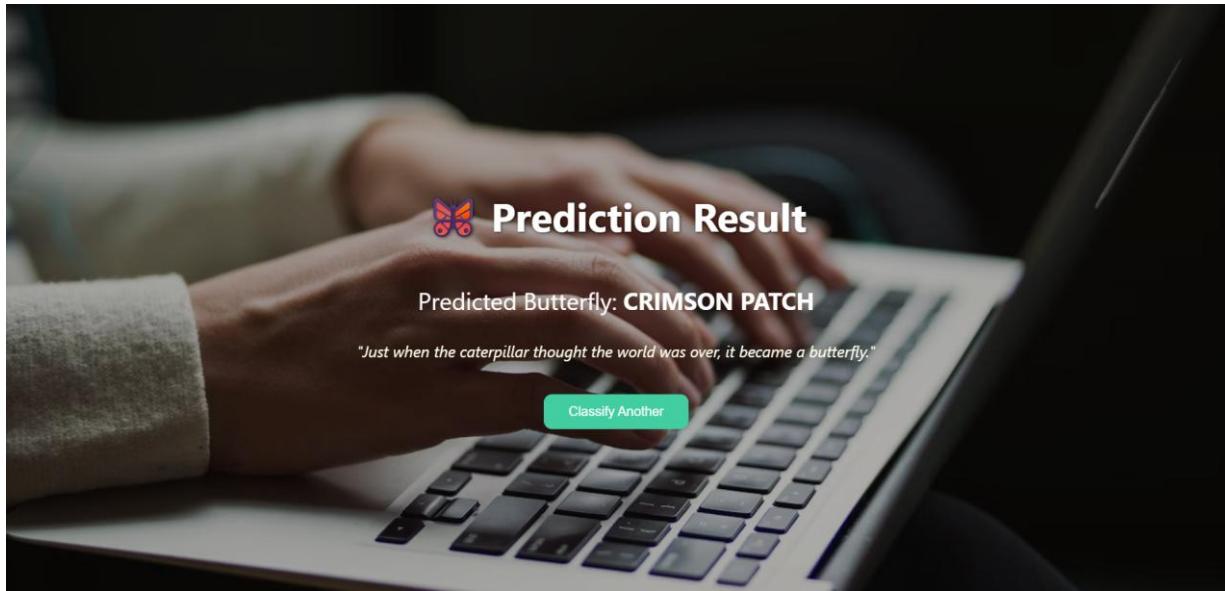
Home page screenshot:



Prediction page screenshot:



Prediction Result screenshot:



10. Known Issues

- **✗** No database – predictions are **not stored** after use
- **⚠** Basic error handling – **input validation** needs enhancement
- **🔓** No authentication – **open access** to the prediction

11. Future Enhancements

- **🔒** Add **doctor login** for secure access
- **💾** Store results in **MongoDB** for record-keeping
- **📄** Enable **PDF report downloads** after prediction
- **🌐** Integrate a **chatbot** or launch a **mobile version** for wider accessibility