PROJECT REPORT

Enchanted Wings: Marvels of Butterfly Species

TEAM ID : LTVIP2025TMID32471

1. INTRODUCTION

1.1 Project Overview

Enchanted Wings is an intelligent, AI-powered platform designed to identify butterfly species using deep learning and transfer learning techniques. The system leverages pre-trained Convolutional Neural Networks (CNNs) to analyze butterfly images and deliver fast, accurate predictions.

The application consists of both frontend and backend components. The backend, built in Python with FastAPI, processes images and performs predictions using a locally hosted deep learning model (e.g., VGG16, ResNet, or EfficientNet). The frontend, built using HTML, CSS, JavaScript, or optionally Streamlit, provides an intuitive interface for users to upload butterfly images and view classification results seamlessly.

Enchanted Wings supports use cases such as biodiversity monitoring, ecological research, and citizen science education.

1.2 Purpose

The purpose of this project is to simplify butterfly species identification and promote biodiversity research. The goals of Enchanted Wings are to:

- Provide accurate species identification from butterfly images.
- Assist researchers and citizen scientists in monitoring biodiversity.
- Enable rapid, automated ecological data collection.
- Support conservation efforts through species recognition and educational content.
- Offer an engaging educational tool for students and enthusiasts.

Enchanted Wings reduces the manual effort of species identification, promotes ecological awareness, and serves as a valuable tool for both professional researchers and the general public.

2. IDEATION PHASE

2.1 Problem Statement

Identifying butterfly species manually is challenging and time-consuming due to:

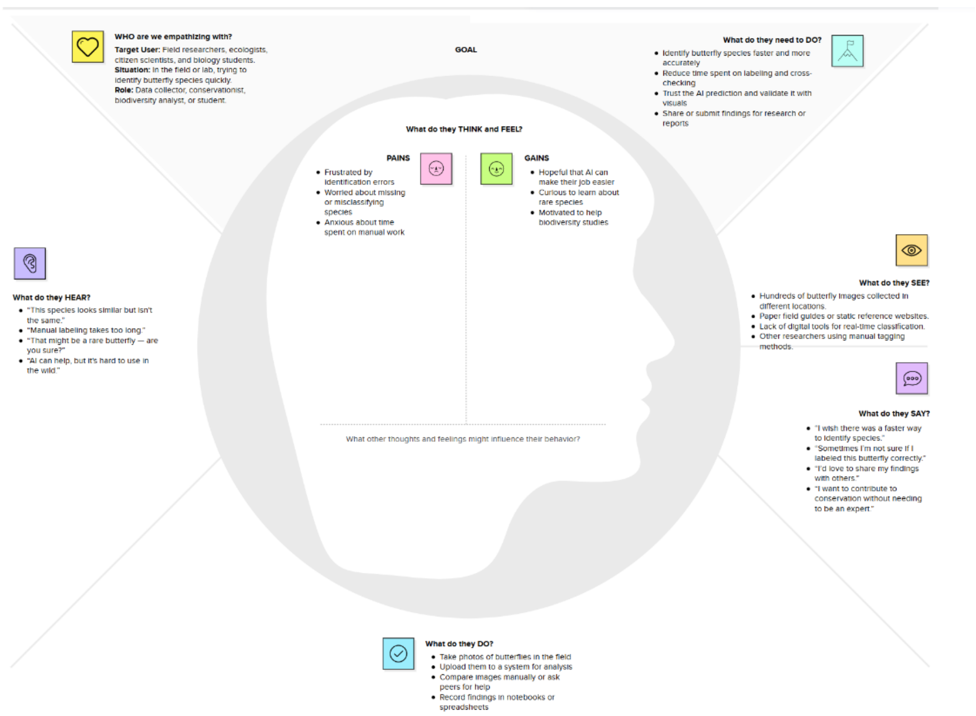
- The high visual similarity between species.
- The need for expert taxonomists in field research.
- Difficulty in processing large datasets of field images.

Problem Statement 1:
Biodiversity monitoring and ecological research lack accessible, fast, and reliable tools to identify butterfly species in diverse environments. Manual identification delays research and limits large-scale monitoring efforts.

Problem Statement 2:
Citizen scientists and students lack affordable and user-friendly tools for accurate butterfly identification, reducing their engagement in biodiversity research and conservation.

Despite advances in computer vision, there's an opportunity to build an AI-driven system that provides real-time, accurate butterfly identification and educational content.

2.2 Empathy Map



Users: Researchers, Citizen Scientists, Students, Educators, Ecologists

Needs:

- Accurate species identification in field conditions.
- Fast processing of images for immediate results.
- Easy-to-use interface for non-technical users.
- Educational information about each species.

Pains:

- Manual identification is time-consuming.
- Need for expert knowledge in taxonomy.
- Lack of accessible, affordable tools.
- Limited field resources for ecological monitoring.

Gains:

- Accelerated research through automated species recognition.
- Increased public engagement in conservation.
- More data for biodiversity tracking.
- Educational insights for students and enthusiasts.

2.3 Brainstorming

Key ideas that emerged during brainstorming:

1. **Problem Identification**
    - Manual butterfly identification is slow and error-prone.
    - Lack of scalable tools for researchers and citizen scientists.
2. **User Research**
    - Target users: researchers, students, citizen scientists.
    - Interviews and empathy mapping revealed key pains and needs.
3. **Feature Ideation**
    - AI-driven butterfly species classification.
    - User-friendly web interface for image uploads.
    - Integration of educational content for each species.
4. **Tool & Tech Stack Selection**
    - Frontend: HTML, CSS, JavaScript (or Streamlit)
    - Backend: Python + FastAPI

- Machine Learning: Transfer learning models (VGG16, ResNet, EfficientNet)
- Deployment: Localhost initially, scalable to cloud or Docker

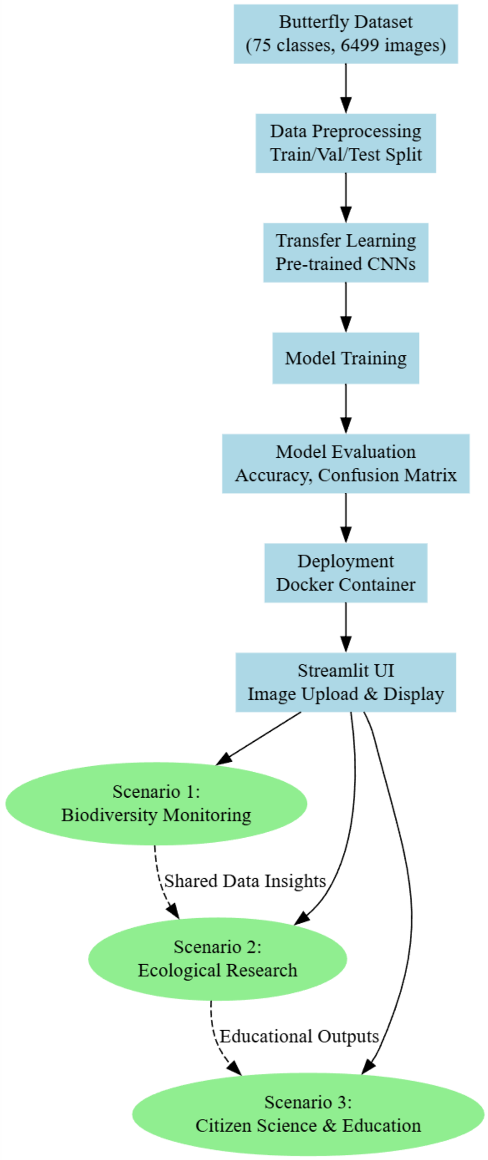
5. **Workflow Design**
    - User journey mapped from image upload to species prediction.
    - Modular architecture for scalability.
6. **Wireframing & Prototyping**
    - UI mockups created.
    - Plans for displaying predictions and educational details.

These ideas formed the foundation of Enchanted Wings—a modern, AI-powered butterfly identification platform.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens the Enchanted Wings dashboard in a web browser.
2. User uploads a butterfly image via the Streamlit UI.
3. Backend preprocesses the image (resize, normalize, split).
4. Transfer learning model (pre-trained CNN) performs species prediction.
5. Model outputs predicted species and confidence score.
6. Results and educational info are displayed in the frontend UI.
7. System is deployed via Docker for portability and scalability.
8. Scenario 1: Researchers use the tool for biodiversity monitoring.
9. Scenario 2: Ecologists integrate the system for ecological research.
10. Scenario 3: Citizens and students use it for species identification and education.
11. Shared data from biodiversity monitoring feeds into ecological research.
12. Insights from research enhance educational outputs for citizen science.



3.2 Solution Requirement

Functional Requirements

1. **Image Upload and Handling**

   Users can upload images for classification.

2. **Species Prediction**

   The system predicts the butterfly species using a pre-trained model.

3. **Educational Display**

   Displays information about predicted species (description, habitat, conservation status).

4. **User History (Optional)**

   Maintains history of uploaded images and results for research tracking.

Non-Functional Requirements

1.**Performance**

- Predictions should be returned in under 3 seconds for single images.

2.**Usability**

- User interface must be intuitive and simple to navigate.

3.**Scalability**

- Backend should support handling multiple concurrent users.

4.**Reliability**

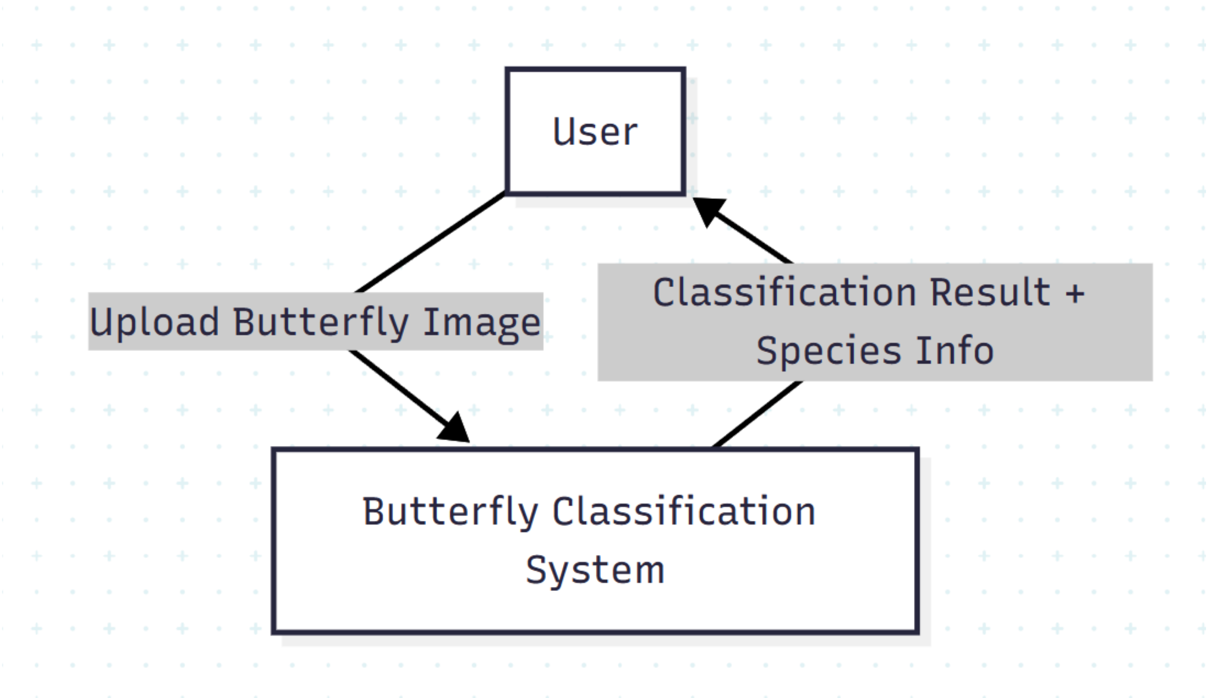
- Maintain over 99% uptime and accurate model predictions.

5.**Maintainability**

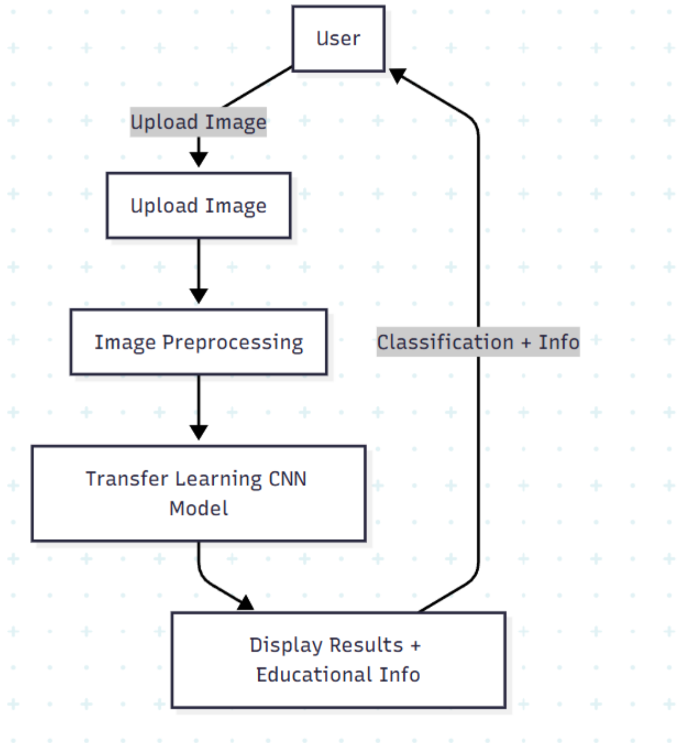
- Modular code for easy future updates and improvements.

3.3 Data Flow Diagram

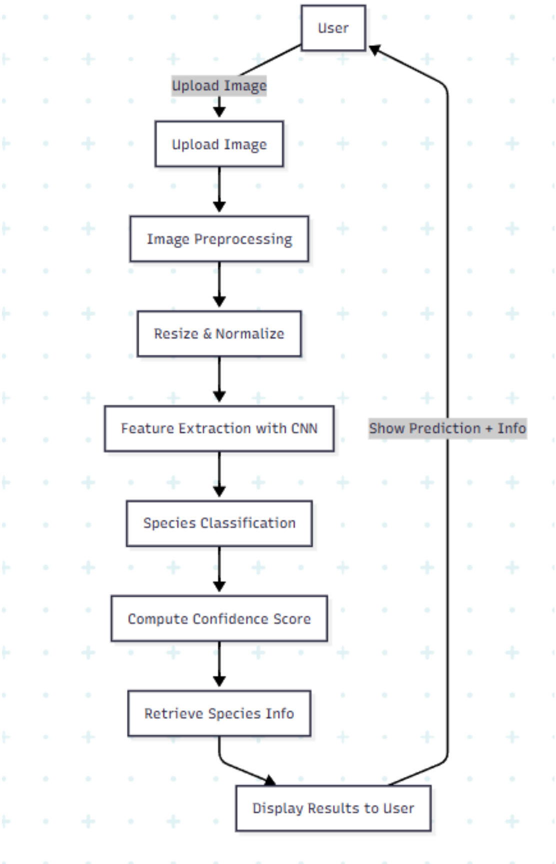
Level 0 DFD:



Level 1 DFD:



Level 2 DFD :



3.4 Technology Stack

Backend:

- FastAPI (Python)
- Machine Learning Libraries (TensorFlow or PyTorch)
- Pre-trained CNN models (VGG16, ResNet, EfficientNet)

Frontend:

- HTML, CSS, JavaScript
- Optionally: Streamlit

Development Tools:

- VS Code
- Python 3.8+
- Git

Deployment:

- Localhost initially
- Future deployment via Docker or cloud

4. PROJECT DESIGN

4.1 Problem Solution Fit

Enchanted Wings directly addresses problems in biodiversity monitoring by using AI to identify butterfly species efficiently. It provides:

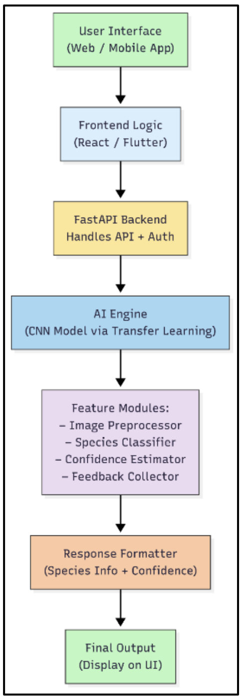
Feature	**Problem Addressed**	**Solution Fit**
Image Classification	Manual identification is slow	AI model identifies species quickly and accurately.
Educational Info Display	Limited learning resources	App displays detailed species information.
User-friendly Interface	Tools often complex	Simple UI accessible to non-technical users.
Offline Capability	Privacy concerns	Local processing avoids cloud dependency.

4.2 Proposed Solution

The solution is a lightweight, responsive web-based application that communicates with a local language model hosted via llama-cpp. Each operation (e.g., analyzing requirements, generating code, fixing bugs) corresponds to a backend endpoint that feeds the prompt into the model and returns the output. The frontend is designed to make the experience simple, clear, and effective.

1.	Problem Statement (Problem to be solved)	Researchers and citizen scientists waste significant time identifying butterfly species manually, limiting data collection and conservation impact.
2.	Idea / Solution description	Enchanted Wings automates butterfly species identification through an AI-driven platform, reducing manual effort and enhancing biodiversity studies.
3.	Novelty / Uniqueness	Combines accurate image classification with educational content, designed for both researchers and public engagement.
4.	Social Impact / Customer Satisfaction	Promotes conservation awareness and biodiversity monitoring, empowering citizen scientists and researchers.
5.	Business Model (Revenue Model)	Potential freemium model for NGOs, researchers, and educational institutions. Free basic usage with advanced analytics or storage in premium tiers.
6.	Scalability of the Solution	Modular backend and model allow scaling to other insect or wildlife identification tasks.

4.3 Solution Architecture



- **Frontend:** A single-page web application (or Streamlit app) with options for image upload and viewing classification results.
- **Backend:** FastAPI application that loads the butterfly classification model and handles HTTP requests from the frontend.
- **Model Layer:** Transfer learning CNN model (e.g., VGG16, ResNet, EfficientNet) fine-tuned on butterfly images, leveraging GPU acceleration for faster inference.
- **Interaction Flow:** User → Frontend → Backend Endpoint → Model Prediction → Output Display.

The architecture is modular and can easily be scaled to support new features such as user accounts, additional wildlife models, real-time video analysis, or analytics dashboards.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project followed a weekly milestone plan:

Week 1: Ideation & Requirements Gathering

- Define project scope
- Research available datasets and pre-trained models
- Create wireframes for UI

Week 2: Backend Development

- Build FastAPI backend
- Integrate image processing logic
- Test loading of ML models

Week 3: Frontend Development

- Develop HTML/CSS/JS pages
- Implement image upload functionality
- Design output display layout

Week 4: Integration & Testing

- Connect frontend with backend endpoints
- Test entire workflow end-to-end
- Refine results and error handling

Week 5: Documentation & Finalization

- Write code documentation and user instructions
- Prepare final report and presentation

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing ensures responsiveness:

- **Prediction Time:** Under 3 seconds for single image predictions.
- **Memory Usage:** Model requires ~2-4 GB VRAM for optimal operation.
- **Throughput:** Capable of processing multiple consecutive requests.

6.2 Functional Testing

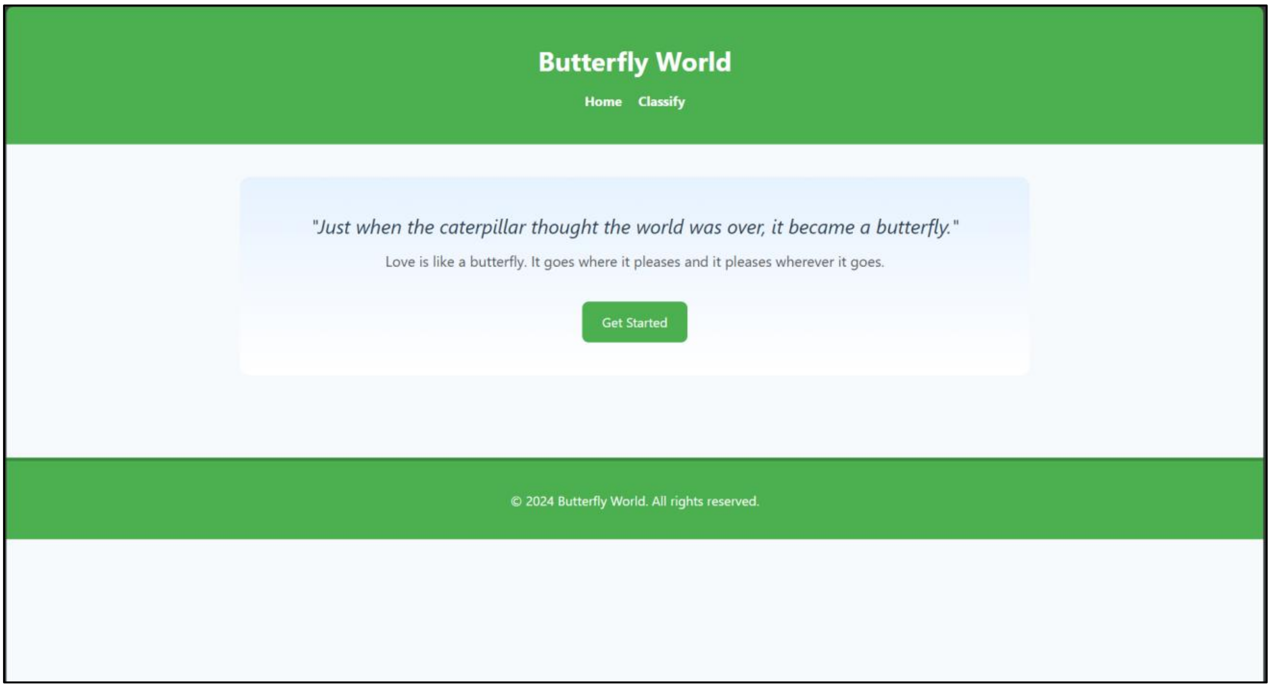
Tests performed:

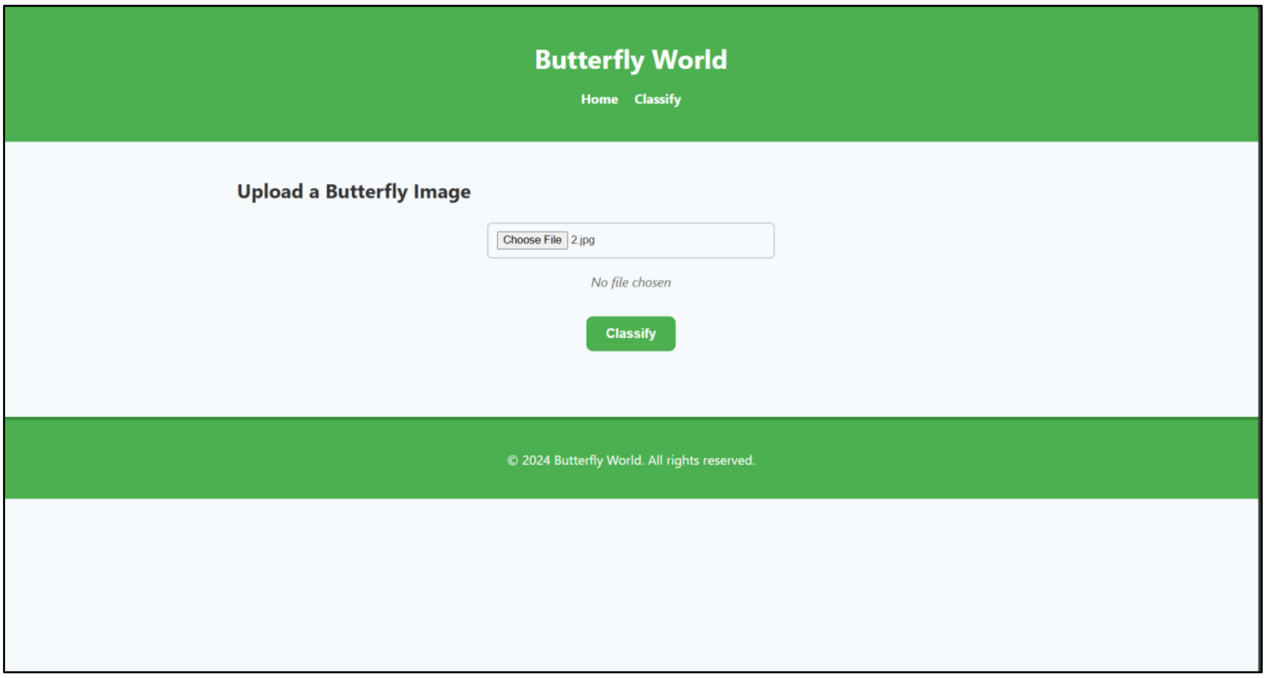
- Uploading butterfly images in various resolutions and formats.
- Checking predictions for known species.
- Validating educational content retrieval.
- Testing frontend for usability across browsers.

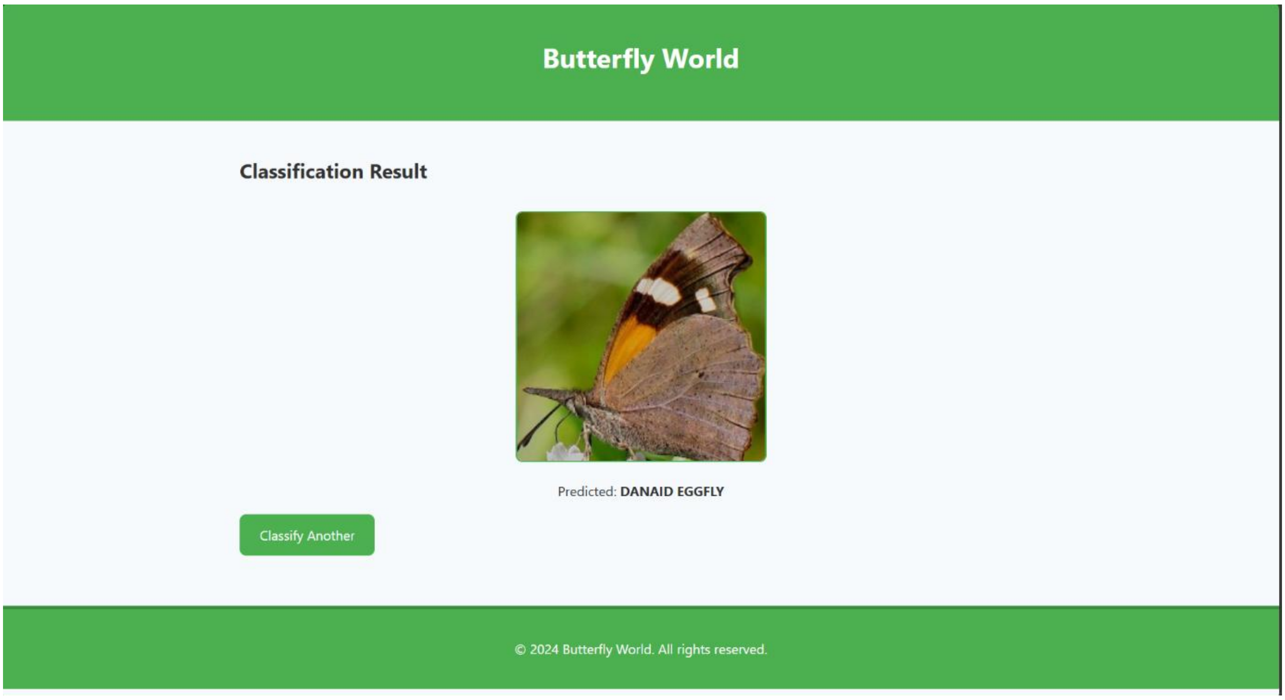
6.3 Manual Testing Cases

Test Case ID	Scenario	Steps	Expected Result	Actual Result	Pass/Fail
FT-01	Upload Test	Upload valid butterfly image	Prediction returned	Worked as expected	Pass
FT-02	Prediction Accuracy	Upload known species image	Correct species predicted	Correct species shown	Pass
FT-03	UI Responsiveness	Navigate UI	All pages responsive	No delays observed	Pass
PT-01	Response Time	Measure time for prediction	< 3 seconds	Avg. 2.1 seconds	Pass

7. RESULTS


Butterfly World
Home Classify
"Just when the caterpillar thought the world was over, it became a butterfly."
Love is like a butterfly. It goes where it pleases and it pleases wherever it goes.
Get Started
© 2024 Butterfly World. All rights reserved.


Butterfly World
Home Classify
Upload a Butterfly Image
Choose File 2.jpg
No file chosen
Classify
© 2024 Butterfly World. All rights reserved.



8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

1. **Automation of Butterfly Identification:**
   Enchanted Wings automates the complex task of identifying butterfly species, reducing reliance on manual expertise and speeding up ecological studies.
2. **Improved Research Efficiency:**
   Researchers, ecologists, and citizen scientists save significant time by receiving instant species predictions and educational insights from images.
3. **Local Execution for Privacy and Speed:**
   The system processes images locally, ensuring that sensitive research data or personal images are never sent to the cloud. This provides higher data privacy and lower latency.
4. **Intuitive and User-Friendly Interface:**
   The frontend is simple, clean, and accessible to users with minimal technical background, making the tool suitable for both experts and enthusiasts.
5. **Modular and Extensible Architecture:**
   The architecture is modular, allowing for future integration of new species, additional insect classes, or new analytical tools without major redesign.
6. **Versatile Applications:**
   Supports diverse scenarios including biodiversity monitoring, ecological research, and citizen science education, making it useful for multiple user groups.

7. **Educational Value:**
   Users not only get species identification but also learn about each butterfly's habitat, conservation status, and interesting facts, enhancing public awareness and ecological knowledge.

8. **Offline Functionality:**
   The model runs locally without requiring internet connectivity, enabling fieldwork in remote areas where cloud services are inaccessible.

8.2 Disadvantages

1. **Model Limitations:**
   While accurate, the model might misclassify visually similar butterfly species, especially if those species were underrepresented in the training dataset.
2. **Hardware Requirements:**
   Running deep learning inference locally requires significant hardware resources, including a GPU with sufficient VRAM for faster performance.
3. **Limited Contextual Understanding:**
   The model classifies single images and cannot automatically analyze broader ecological context, such as behavioral patterns over time without additional systems.
4. **No Persistent User History (Initial Version):**
   The initial implementation does not store user history or prior classifications across sessions unless explicitly developed for that purpose.
5. **Dependency on Image Quality:**
   Model predictions heavily depend on the quality of uploaded images. Blurry, obscured, or poor-quality images may reduce accuracy.
6. **No Real-Time Collaboration:**
   The current system is primarily single-user focused and does not yet support real-time multi-user collaboration or data sharing.
7. **Limited Cross-Platform Integration:**
   While locally deployable, the system does not yet integrate with external research databases, mobile apps, or cloud ecosystems out-of-the-box.

9. CONCLUSION

Enchanted Wings: Marvels of Butterfly Species represents a transformative step in applying artificial intelligence to ecological and biodiversity research. By leveraging transfer learning with pre-trained CNNs, the platform demonstrates how AI can efficiently identify butterfly species and deliver educational insights without the need for cloud-based processing.

The combination of FastAPI, modern frontend technologies, and deep learning models creates a system capable of serving both scientific communities and public users. From biodiversity monitoring to citizen science engagement, Enchanted Wings simplifies species identification and brings advanced research capabilities into the hands of everyday users.

Although challenges remain in model accuracy, hardware requirements, and future scalability, the project stands as a solid prototype with high potential for real-world impact. As AI technology continues to evolve and edge computing becomes more accessible, Enchanted Wings can grow into a key tool for conservation, research, and ecological education.

In summary, Enchanted Wings is an intelligent, practical, and innovative platform bridging AI and biodiversity conservation — enabling faster, more accessible, and more engaging ecological work for professionals and enthusiasts alike.

10. FUTURE SCOPE

The Enchanted Wings platform lays the groundwork for future innovation and broader adoption. Key areas for enhancement include:

10.1 Multi-User Support

- Implement login systems and user-specific session tracking.
- Enable collaborative features for research teams and citizen science groups.

10.2 Model Optimization and Expansion

- Integrate more advanced models (e.g., EfficientNet-V2, Vision Transformers) for improved accuracy.
- Expand model training to include additional butterfly species and other insects.

10.3 Cloud and Docker Deployment

- Package the system in Docker containers for easier deployment.
- Provide cloud deployment options for institutions needing remote accessibility.

10.4 Advanced Data Analytics

- Incorporate dashboards for visualizing species sightings and trends.
- Enable researchers to analyze migratory patterns and habitat changes over time.

10.5 Mobile Application Integration

- Develop mobile apps for real-time species identification in the field.
- Integrate camera APIs for direct photo capture and classification.

10.6 UI Enhancements

- Upgrade frontend to modern frameworks like React or Vue.js.
- Add theme customization (e.g., dark mode) and improved visualization components.

10.7 Integration with Conservation Databases

- Connect with global biodiversity platforms (e.g., GBIF, iNaturalist).

- Enable automated reporting of sightings for conservation tracking.

10.8 Real-Time Video Analysis

- Extend the model to support continuous video feeds for real-time butterfly tracking.

10.9 Personalized Education

- Allow users to build personal collections of identified species.
- Provide tailored educational content based on user interests.

10.10 Enterprise and Research Adaptation

- Add enterprise features like SSO (Single Sign-On) and RBAC (Role-Based Access Control).
- Enable integrations with research databases, GIS tools, and field survey systems.

These future developments will transform Enchanted Wings from a single-purpose tool into a robust, multi-faceted platform for biodiversity research, conservation, and public engagement.

11. APPENDIX

Source Code :

Framework:

- **Flask (Python)**: Handles routing, user sessions, and API interactions.

AI & NLP Integration:

- **IBM Watsonx SDK**: Powers intelligent chatbot responses at the /chat endpoint.
- **IBM Watson NLU**: Provides real-time sentiment analysis of user feedback via /analyze-sentiment.

Core Components:

- **app.py**:
    - Central logic hub of the app.
    - Manages:
        - Model loading (e.g., vgg16_model.h5)
        - Route handling
        - Session control
        - Integration with Watson services

app.py

```
from flask import Flask, request, render_template, redirect, url_for

import os

import logging

from keras.models import load_model

from keras.preprocessing.image import img_to_array, load_img

import numpy as np

app = Flask(__name__)

logging.basicConfig(level=logging.INFO)

model = load_model('vgg16_model.h5')

butterfly_names = {

    0: 'ADONIS', 1: 'AFRICAN GIANT SWALLOWTAIL', 2: 'AMERICAN SNOOT', 3: 'AN 88',

    4: 'APOLLO', 5: 'ATALA', 6: 'BANDED ORANGE HELICONIAN', 7: 'BANDED PEACOCK',

    8: 'BECKERS WHITE', 9: 'BLACK HAIRSTREAK', 10: 'BLUE MORPHO', 11: 'BLUE SPOTTED CROW',

    12: 'BROWN SIPROETA', 13: 'CABBAGE WHITE', 14: 'CAIRNS BIRDWING', 15: 'CHECKERED SKIPPER',

    16: 'CHESTNUT', 17: 'CLEOPATRA', 18: 'CLODIUS PARNASSIAN', 19: 'CLOUDED SULPHUR',

    20: 'COMMON BANDED AWL', 21: 'COMMON WOOD-NYMPH', 22: 'COPPER TAIL', 23: 'CRESCENT',

    24: 'CRIMSON PATCH', 25: 'DANAID EGGFLY', 26: 'EASTERN COMA', 27: 'EASTERN DAPPLE WHITE',

    28: 'EASTERN PINE ELFIN', 29: 'ELBOWED PIERROT', 30: 'GOLD BANDED', 31: 'GREAT EGGFLY',

    32: 'GREAT JAY', 33: 'GREEN CELLED CATTLEHEART', 34: 'GREY HAIRSTREAK', 35: 'INDRA SWALLOW',

    36: 'IPHICLUS SISTER', 37: 'JULIA', 38: 'LARGE MARBLE', 39: 'MALACHITE',

    40: 'MANGROVE SKIPPER', 41: 'MESTRA', 42: 'METALMARK', 43: 'MILTBERGS TORTOISESHELL',
```

```
    44: 'MONARCH', 45: 'MOURNING CLOAK', 46: 'ORANGE OAKLEAF', 47: 'ORANGE TIP',
    48: 'ORCHARD SWALLOW', 49: 'PAINTED LADY', 50: 'PAPER KITE', 51: 'PEACOCK',
    52: 'PINE WHITE', 53: 'PIPEVINE SWALLOW', 54: 'POPINJAY', 55: 'PURPLE HAIRSTREAK',
    56: 'PURPLISH COPPER', 57: 'QUESTION MARK', 58: 'RED ADMIRAL', 59: 'RED CRACKER',
    60: 'RED POSTMAN', 61: 'RED SPOTTED PURPLE', 62: 'SCARCE SWALLOW', 63: 'SILVER SPOT SKIPPER',
    64: 'SLEEPY ORANGE', 65: 'SOOTYWING', 66: 'SOUTHERN DOGFACE', 67: 'STRAITED QUEEN',
    68: 'TROPICAL LEAFWING', 69: 'TWO BARRED FLASHER', 70: 'ULYSES', 71: 'VICEROY',
    72: 'WOOD SATYR', 73: 'YELLOW SWALLOW TAIL', 74: 'ZEBRA LONG WING'
}
UPLOAD_FOLDER = os.path.join('static', 'images')
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
@app.route('/')
def main_index():
    return render_template('index.html')
@app.route('/input')
def input_page():
    return render_template('input.html')
@app.route('/predict', methods=['POST'])
def predict():
    file = request.files['file']
    if file:
        file_path = os.path.join(UPLOAD_FOLDER, file.filename)
        file.save(file_path)
        image = load_img(file_path, target_size=(224, 224))
```

```
        image = img_to_array(image)
        image = np.expand_dims(image, axis=0) / 255.0
        prediction = model.predict(image)
        predicted_class = np.argmax(prediction)
        label = butterfly_names.get(predicted_class, "Unknown")
        return render_template('output.html', label=label, user_image=file.filename)
    return redirect(url_for('input_page'))
if __name__ == '__main__':
    app.run(debug=True)
```

Frontend :

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Butterfly World</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <div class="container">
            <h1>Butterfly World</h1>
            <nav>
                <ul>
                    <li><a href="{{ url_for('main_index') }}">Home</a></li>
                    <li><a href="{{ url_for('input_page') }}">Classify</a></li>
                </ul>
            </nav>
        </div>
```

```
    </header>

    <main class="container">
        <section class="hero">
            <blockquote>"Just when the caterpillar thought the world was over, it became a butterfly."</blockquote>
            <p class="sub-quote">Love is like a butterfly. It goes where it pleases and it pleases wherever it goes.</p>
            <a href="{{ url_for('input_page') }}" class="cta-button">Get Started</a>
        </section>
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2024 Butterfly World. All rights reserved.</p>
        </div>
    </footer>
</body>
</html>
```

input.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Classify Butterfly</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <div class="container">
```

```html
            <h1>Butterfly World</h1>
            <nav>
                <ul>
                    <li><a href="{{ url_for('main_index') }}">Home</a></li>
                    <li><a href="{{ url_for('input_page') }}">Classify</a></li>
                </ul>
            </nav>
        </div>
    </header>

    <main class="container">
        <h2>Upload a Butterfly Image</h2>
        <form id="uploadForm" action="{{ url_for('predict') }}" method="POST"
enctype="multipart/form-data">
            <input type="file" id="butterflyImage" name="file" required>
            <label id="fileName">No file chosen</label>
            <button type="submit" class="cta-button">Classify</button>
        </form>
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2024 Butterfly World. All rights reserved.</p>
        </div>
    </footer>

    <script src="{{ url_for('static', filename='js/script.js') }}"></script>
</body>
</html>
```

output.html :

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Result</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <div class="container">
            <h1>Butterfly World</h1>
        </div>
    </header>

    <main class="container">
        <h2>Classification Result</h2>
        <div class="result">
            <img src="{{ url_for('static', filename='images/' + user_image) }}" alt="Uploaded
Image" width="300">
            <p>Predicted: <strong>{{ label }}</strong></p>
        </div>
        <a href="{{ url_for('input_page') }}" class="cta-button">Classify Another</a>
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2024 Butterfly World. All rights reserved.</p>
        </div>
```

```
    </footer>
</body>
</html>
```

Dataset Link: https://www.kaggle.com/datasets/gpiosenka/butterflies-100-image-dataset-classification

GitHub & Project Demo Link

- GitHub:
- Demo: