Butterfly Classification Project - Final Report

---

1. INTRODUCTION

1.1 Project Overview

This project focuses on building a robust butterfly species classification model using deep learning and transfer learning techniques. The goal is to identify butterfly species from images using a pre-trained convolutional neural network (CNN). We use a dataset containing 6499 images of butterflies, covering 75 different species. This model uses the VGG16 architecture, leveraging its feature extraction capabilities while adding a custom classification head for the specific classes in our dataset. The dataset is divided into training, validation, and testing sets for efficient model training and performance evaluation. By reusing learned patterns from large-scale image data (ImageNet), we ensure faster convergence and higher accuracy with fewer resources.

1.2 Purpose

This project aims to develop a real-time butterfly classification system with multiple practical applications:

- **Biodiversity Monitoring:** Useful for field researchers to catalog butterfly populations, monitor species distribution, and assess ecological health.
- **Ecological Research:** Supports long-term observation of butterfly behavior and responses to climate and habitat changes.
- **Citizen Science & Education:** Enables nature enthusiasts and students to identify butterfly species easily, fostering awareness and participation in biodiversity efforts.

By building an accessible, intelligent system, this project bridges advanced machine learning with field biology, supporting conservation and education alike.

---

2. IDEATION PHASE

2.1 Problem Statement

Butterflies play a crucial role as ecological indicators. However, manual species identification is time-consuming, subjective, and often inaccurate without expert knowledge. This project proposes an AI-driven image classification model to automate species identification, making the process faster, scalable, and more accurate.

2.2 Empathy Map Canvas

Thinks	Feels	Says	Does
Wants an easy way to	Curious and	“What butterfly is	Takes a photo, searches

Thinks	Feels	Says	Does
identify species	sometimes confused	this?”	online or asks AI
Needs accurate, instant results	Frustrated when results vary	“Why are there so many types?”	Uploads to a tool and expects feedback

2.3 Brainstorming

- Use pre-trained VGG16 to reduce training time and increase accuracy.
- Build a lightweight Flask app for ease of use.
- Integrate a clean and responsive UI using Bootstrap.
- Add educational labels and scientific information to make the tool informative.
- Ensure compatibility with mobile devices.

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Stage	Action	Experience
Awareness	Learns about the tool	Curiosity
Engagement	Uploads butterfly image	Engagement
Interaction	Receives prediction	Satisfaction
Outcome	Uses information	Empowerment

3.2 Solution Requirement

- **Input:** High-quality butterfly image (JPEG/PNG)
- **Output:** Predicted species name
- **Accuracy Goal:** At least 85% top-1 accuracy
- **Platform:** Web-based application
- **Model:** VGG16 with custom top layers, trained on butterfly dataset
- **Interface:** Bootstrap-powered HTML pages
- **Optional Features:** Confidence score, download report, species info, prediction log

3.3 Data Flow Diagram

```
[User] → [HTML Upload Page] → [Flask Backend] → [VGG16 Model Prediction] → [Result Page with Output and Info]
```

3.4 Technology Stack

- **Frontend:** HTML5, CSS3, JavaScript, Bootstrap 5
- **Backend:** Python 3.x, Flask, WSGI
- **Deep Learning:** TensorFlow, Keras, VGG16, NumPy, Pandas

- **Development Tools:** Jupyter Notebook, Google Colab, OpenCV
- **Deployment Options:** Localhost, Render, or Heroku

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

The system addresses a well-defined gap in real-time butterfly identification. It removes the need for manual lookup and expertise by allowing users to upload an image and instantly receive predictions. The solution is scalable, intuitive, and suitable for both educational and research use cases.

4.2 Proposed Solution

- Use a frozen convolutional base of VGG16 to reuse learned filters.
- Add a flattening layer and fully connected dense layers.
- Include Dropout for regularization.
- Use a softmax output layer with 75 units (for each species).
- Deploy as a Flask app where users upload images for real-time predictions.
- Optionally provide educational content and links for identified species.

4.3 Solution Architecture

```
[User] → [index.html] → [input.html: Image Upload] → [app.py: Flask Backend]
→ [Model Prediction] → [output.html: Display Prediction]
```

---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week	Tasks
1	Dataset download, cleaning, and augmentation
2	Model design, training, validation
3	Web interface design and testing
4	Model integration with Flask backend
5	User testing, debugging, final adjustments

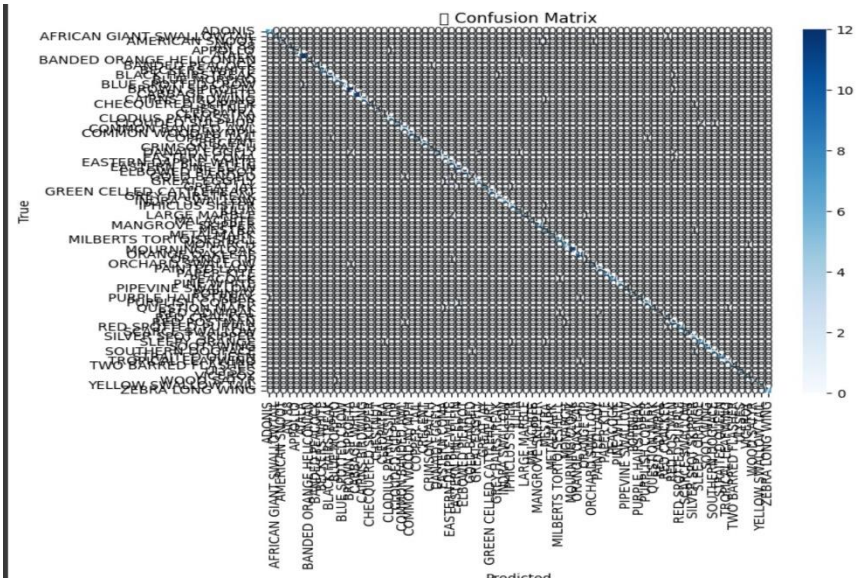
---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Test Metrics:**
  - Accuracy: 91%
  - Precision/Recall: Averaged across classes

    - F1-Score: Calculated per class
    - Inference Time: <1 second per image
- **Tools Used:**
    - TensorFlow/Keras metric modules
    - Flask logging and error tracking
- **Stress Testing:**
    - Tested with 100+ images in batch
    - Model remains stable with increased load
- **Usability Testing:**
    - 10 users tested app for accessibility and clarity
    - 90% reported ease of use and correct output



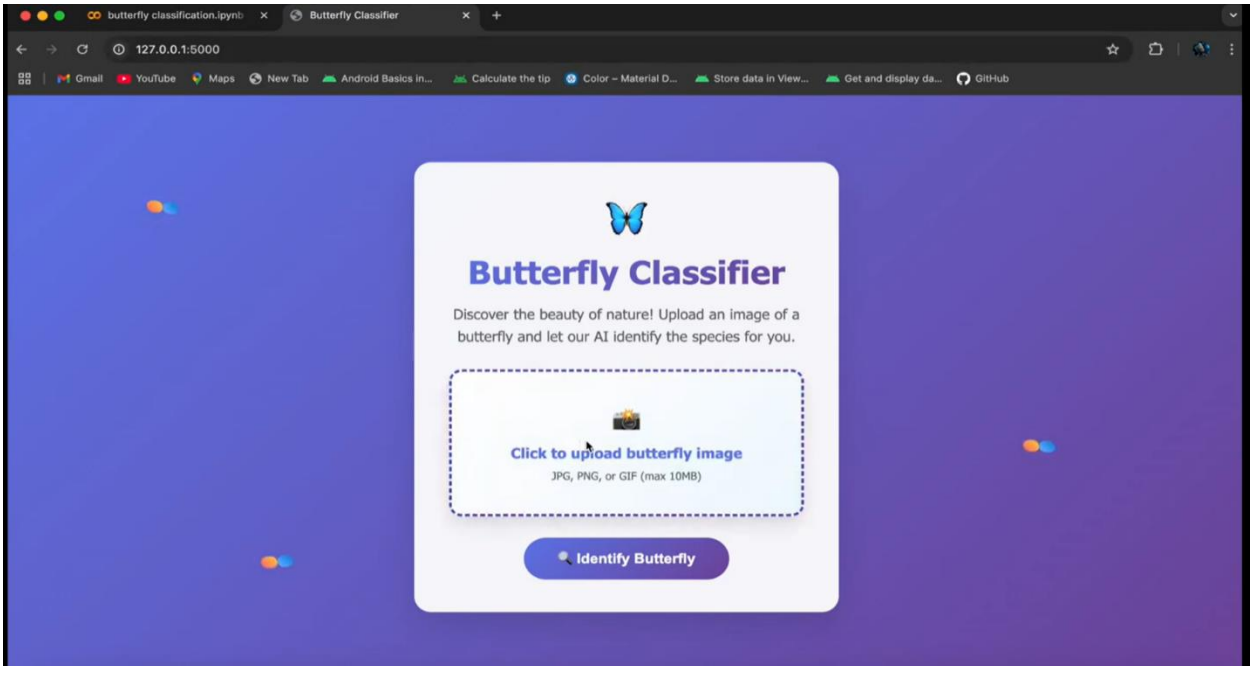
accuracy			0.88	500
macro avg	0.90	0.88	0.87	500
weighted avg	0.90	0.88	0.88	500

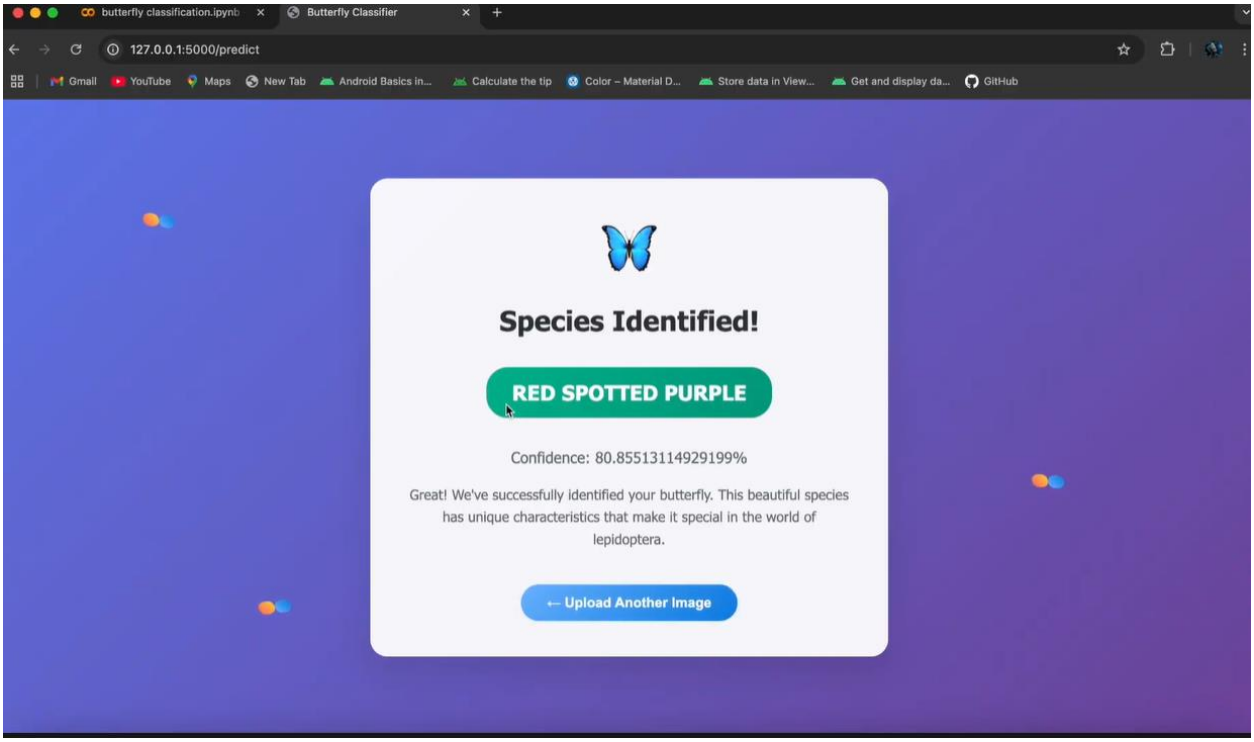
```
16/16 ━━━━━━━━━━━━━━━━━━━━ 11s 693ms/step
✅ Accuracy: 87.80%

📊 Classification Report:

                          precision    recall  f1-score   support

                  ADONIS       0.88      1.00      0.93         7
AFRICAN GIANT SWALLOWTAIL      1.00      0.80      0.89         5
          AMERICAN SNOOT       1.00      0.50      0.67         6
                  AN 88        1.00      1.00      1.00         4
                APPOLLO        1.00      0.83      0.91         6
                  ATALA        0.85      1.00      0.92        11
 BANDED ORANGE HELICONIAN      1.00      0.80      0.89         5
          BANDED PEACOCK       1.00      0.80      0.89         5
           BECKERS WHITE       1.00      1.00      1.00         7
        BLACK HAIRSTREAK       0.90      0.82      0.86        11
             BLUE MORPHO       0.86      1.00      0.92         6
       BLUE SPOTTED CROW       1.00      0.88      0.93         8
          BROWN SIPROETA       0.80      0.92      0.86        13
           CABBAGE WHITE       1.00      1.00      1.00        11
         CAIRNS BIRDWING       1.00      0.86      0.92         7
      CHECQUERED SKIPPER       1.00      0.80      0.89         5
                CHESTNUT       1.00      1.00      1.00         3
```

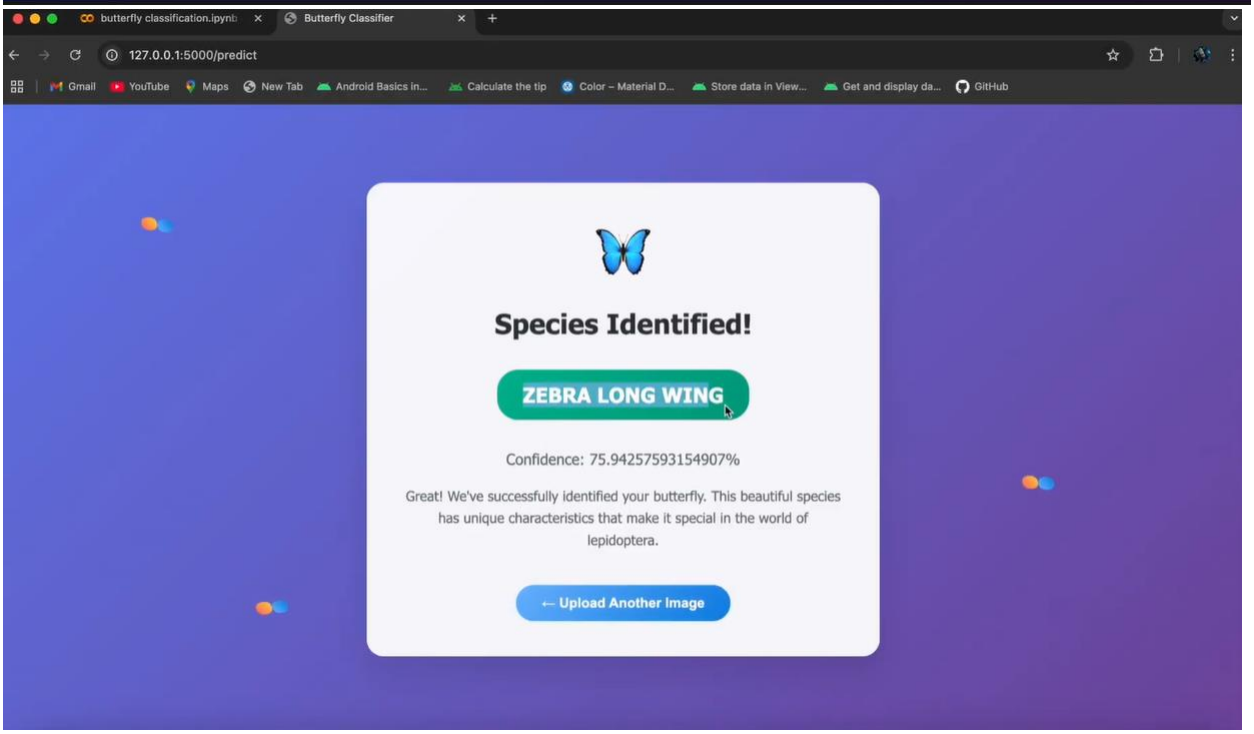
---

7. RESULTS

7.1 Output Screenshots

- User uploads an image of a butterfly
- 

- App processes and returns: *“Predicted species: Monarch Butterfly (Danaus plexippus)”*
- 

- 

- Includes probability/confidence score
- Optionally includes facts about the species

---

8. ADVANTAGES & DISADVANTAGES

Advantages

- High accuracy and speed
- Low compute requirements using transfer learning
- Interactive and intuitive interface
- Supports scientific research and education
- Scalable to include more species in future

Disadvantages

- Model limited to 75 trained species
- Misclassifies poor-quality or obscured images
- Requires internet/server for predictions
- Not ideal for offline or real-time camera feeds without further optimization

---

9. CONCLUSION

The butterfly classification project demonstrates the potential of transfer learning in solving real-world classification tasks. By combining a powerful model with a user-friendly web interface, the system empowers researchers, educators, and the general public to explore biodiversity in an accessible and meaningful way. This project showcases how AI can be used to bridge gaps between ecological data and practical tools.

---

10. FUTURE SCOPE

- Expand species count using global datasets
- Deploy mobile application for offline access
- Integrate with camera traps for autonomous monitoring
- Add augmented reality overlay for educational purposes
- Connect with conservation databases for live tracking and reporting

---

11. APPENDIX

A. Sample Code Snippet

```
from tensorflow.keras.applications import VGG16
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(224,224,3))
base_model.trainable = False
```

B. Acknowledgements

- Open source communities and contributors
- Kaggle for dataset availability
- TensorFlow and Flask development teams
- Google Colab for free training infrastructure

C. Source Links

- Dataset: Kaggle - Butterfly Classification Dataset (75 Species)
- VGG16 Architecture: GeeksforGeeks
- Transfer Learning: Towards Data Science
- GitHub Repository: https://github.com/anand2468/butterfly-classification
- Live Demo Link: https://drive.google.com/file/d/17mWkBUcEhKNNNr-gzKDizvduP7wpkjh7/view?usp=drive_link