Butterfly Species Classification with EfficientNetB0

1. Project Overview

This project focuses on **classifying butterfly species** from images. The dataset contains **40 butterfly species**, each stored in separate folders. The task is a **multi-class image classification** problem.

The chosen model is **EfficientNetB0**, a lightweight yet powerful convolutional neural network. The model was initialized with **ImageNet pre-trained weights** and then **fine-tuned** on the butterfly dataset.

---

2. Workflow

1. **Data Preparation**
   - The dataset was split into train and valid directories.
   - **Data augmentation** (rotation, shifts, zoom, and horizontal flips) was applied to increase data diversity and prevent overfitting.
2. **Model Building**
   - The convolutional base of EfficientNetB0 was loaded without its top layers.
   - New layers were added: **GlobalAveragePooling2D**, a **Dense layer with 128 ReLU units**, and a **softmax output layer** for 40 classes.
   - The model was compiled with **Adam optimizer (learning rate = 1e-4)**, **categorical crossentropy loss**, and **accuracy** as the evaluation metric.
3. **Model Training**
   - The model was trained for **10 epochs** using the prepared training and validation data.
   - The trained model was saved as efficientnet_b0_trained.keras.
4. **Model Evaluation**
   - The script evaluate_model.py was used to load and evaluate the model on test data.
   - Outputs include the **evaluation report** (evaluation_report_2025-05-25_10-24-07.txt) and the **confusion matrix image** (confusion_matrix.png).

- Results showed **98% accuracy** and a **macro F1-score of 0.98**, indicating excellent performance.

---

3. Key Results

- **Accuracy:** 0.98
- **Macro F1-score:** 0.9798
- Most species achieved **precision and recall of 1.00**.
- A few species, such as **GREAT EGGFLY**, had slightly lower recall (0.60), but this had minimal impact on overall performance.

---

4. Project Files

File / Folder	Description
butterfly-images40-species.zip	Dataset containing images of 40 butterfly species (training and validation).
train_model.py	Script for training the EfficientNetB0 model and saving the trained weights.
evaluate_model.py	Script for loading the trained model, evaluating on test data, and reporting results.
evaluation_report_2025-05-25_10-24-07.txt	Evaluation report with accuracy, F1-score, and per-class metrics.
confusion_matrix.png	Confusion matrix image summarizing classification results.
efficientnet_b0_trained.keras	Saved model file after training.
README.md	A concise project description for GitHub.

---

5. Tools and Technologies

- **Programming Language:** Python

- **Libraries:** TensorFlow/Keras (EfficientNetB0, ImageDataGenerator), NumPy, Matplotlib
- **Hardware:** GPU acceleration recommended for faster training

---

6. Conclusions and Recommendations

- EfficientNetB0 demonstrated **excellent performance** in classifying butterfly species.
- To improve recall for underperforming classes, consider:
    - Increasing the number of training epochs or using **Early Stopping** and **Learning Rate Scheduling**.
    - Gathering additional images for classes with fewer samples.
    - Unfreezing and fine-tuning more layers of EfficientNetB0 with a reduced learning rate.

---

7. Potential Applications

- Building lightweight, accurate models for **ecological research** and **species identification**.
- Serving as an educational example for **transfer learning** and **multi-class image classification**.