Intelligence Academy
The Center for Research and Development

🦋 Butterfly Image Classification Capstone Project

Automated Butterfly Species Identification Using Deep Learning

📌 Project Assessment Criteria

Section	Marks (%)	Description
Introduction & Background	10%	Problem statement, objectives, and significance.
Literature Review	10%	Review of existing solutions and deep learning techniques.
Dataset & Preprocessing	15%	Data understanding, augmentation, and preprocessing steps.
Methodology (Model Design)	20%	CNN architectures, training process, and hyperparameter tuning.
Evaluation & Results	15%	Model performance analysis using metrics, visualizations.
Deployment & Web Application	10%	API development and UI implementation for real-world use.
Discussion & Conclusion	10%	Error analysis, limitations, and future improvements.
Report Quality & Presentation	10%	Clarity, formatting, documentation, and references.

📖 1. Introduction

1.1 Background

Butterflies are essential to ecosystems, playing a role in pollination and indicating environmental health. However, manual classification is **challenging** due to:

- **Similar wing patterns** across species.
- **High intra-class variation** due to environmental conditions.
- **Manual expertise requirement** for accurate identification.

1.2 Problem Statement

This project aims to **automate butterfly species identification** using deep learning, particularly Convolutional Neural Networks (CNNs). The goal is to classify **75 different butterfly species** from images.

1.3 Objectives

- Train a deep learning model to **classify butterfly species**.
- Improve classification accuracy using **transfer learning and augmentation**.
- **Deploy the model** as a web-based application for real-time predictions.

1.4 Significance

- **Biodiversity Conservation**: Helps researchers identify rare species.
- **Citizen Science**: Enables public participation in species tracking.
- **Automated Research Tool**: Supports ecological and entomological studies.

📚 2. Literature Review (10%)

2.1 Traditional Approaches

- **Manual identification** by taxonomists.
- **Feature-based ML models** (SVM, k-NN) but require handcrafted features.

2.2 Deep Learning in Image Classification

- **CNNs (Convolutional Neural Networks)** dominate image recognition tasks.
- Pretrained architectures like **ResNet, VGG, and EfficientNet** improve accuracy.
- **Data Augmentation & Fine-Tuning** enhance performance in small datasets.

2.3 Existing Research

- **Insect identification models** have reached **85-90% accuracy** with CNNs.
- Most **lack class diversity** or are **not real-world deployable**.

📊 3. Dataset & Preprocessing (15%)

3.1 Dataset Overview

- **Source**: Dataset consists of **1000+ labeled images** of **75 butterfly species**.
- **Train/Test Split**:
  - Training images: **Training_set.csv**
  - Testing images: **Testing_set.csv** (unlabeled images for prediction).

3.2 Preprocessing Steps

- **Image Resizing**: All images resized to **224x224 pixels** for consistency.
- **Normalization**: Pixel values scaled between **[0,1]**.
- **Data Augmentation**:
  - **Rotation (±20°), Flipping, Brightness Adjustments**.
  - **Synthetic images** generated for underrepresented classes.

3.3 Exploratory Data Analysis

- **Class Distribution Visualization** (Identify data imbalance).
- **Sample Image Grid** (Show representative species).
- **Outlier Detection** (Remove corrupted images).

🧠 4. Methodology (Model Design) (20%)

4.1 Model Selection

We test various **CNN architectures**:

1. **Baseline CNN**: 3 Conv layers, Pooling, Dense layers.
2. **Pretrained Models**:
   - **VGG16**
   - **ResNet-50**
   - **EfficientNetB0**

3. **Fine-tuned Model**:
    - Best architecture is **fine-tuned** using **Transfer Learning**.

4.2 Model Architecture

- **Input Layer**: Accepts 224x224 images.
- **Convolutional Layers**: Extract hierarchical features.
- **Batch Normalization**: Stabilizes training.
- **Dense Layer**: Final classification into **75 classes**.
- **Activation Function**: Softmax for multi-class classification.

4.3 Training Setup

- **Loss Function**: Cross-Entropy Loss.
- **Optimizer**: Adam / SGD with momentum.
- **Learning Rate Schedule**: ReduceLROnPlateau.
- **Hyperparameter Tuning**: Grid search for best dropout rate, batch size.

4.4 Model Training

- **Trained for 50 epochs** with early stopping.
- **GPU-accelerated** training on **Google Colab / Kaggle Notebooks**.

📈 5. Evaluation & Results (15%)

5.1 Performance Metrics

- **Accuracy**: Measures overall correctness.
- **Precision & Recall**: Balance between false positives & false negatives.
- **F1-Score**: Best metric for **imbalanced classes**.

5.2 Confusion Matrix

- **Visualize misclassifications** to identify problematic classes.

5.3 Model Comparison

Model	Accuracy	F1-Score
Baseline CNN	72%	0.70

Model	Accuracy	F1-Score
ResNet-50	85%	0.84
EfficientNetB0	**91%**	**0.90**

5.4 Error Analysis

- Investigate **misclassified images**.
- Study **challenging species** with overlapping patterns.

🌍 6. Deployment & Web Application (10%)

6.1 API Development

- **Framework**: FastAPI / Flask.
- **Model hosted** on Hugging Face Spaces / Google Cloud.

6.2 Web Interface

- **Built with Streamlit** for an interactive UI.
- **Features**:
  - Upload an image.
  - Get real-time butterfly species prediction.
  - Display top-5 similar species.

6.3 Docker & Cloud Hosting (Optional)

- Containerized using **Docker**.
- **Deployment on AWS/GCP** for accessibility.

📢 7. Discussion & Conclusion (10%)

7.1 Key Findings

- **Achieved 91% accuracy** using EfficientNet.
- **Fine-tuning improved generalization**.

7.2 Limitations

- Small dataset; **could benefit from synthetic data**.
- **No multi-angle images** (only frontal views).

7.3 Future Improvements

- **Use Self-Supervised Learning** for better feature extraction.
- **Mobile App Deployment** for field research.

📚 8. References (10%)

- Research papers on **CNN-based species classification**.
- Official documentation of **TensorFlow, PyTorch**.
- Biodiversity studies on butterfly species.

📝 Final Submission Checklist

✅ **Code Repository** (GitHub link).
✅ **Final Report** (PDF).
✅ **Web App Link** (Deployed Model).
✅ **Presentation Slides** (Capstone Defense).