Butterfly Image Classification — Project Documentation

Introduction:

Project Title:

Butterfly Classification: Butterfly Species Classification Using Deep Learning and Web Integration

Team Members :

Yadavakunta Pradeep Reddy

Vijayakumar Poojitha

G Pranay Kumar

Laisetty Prem Kumar

Project Overview:

This project is a deep learning-based web system that classifies butterflies into 75 species using image uploads. A trained **VGG16 Convolutional Neural Network (CNN)** powers the model prediction, and a **Flask** backend handles routing, inference, and serving templates. The project is fully responsive with a clean, modern UI for both desktop and mobile users.

Project Goal

- Build an image classifier to distinguish between various butterfly species using deep learning.
- Provide an end-to-end, user-friendly web interface to interact with the model.
- Serve as a reliable tool for ecological research, biodiversity cataloging, and education.

Applications:

- **Ecological Research**: Assists researchers in identifying butterfly species in the wild.
- **Conservation Efforts**: Helps track and monitor endangered species.
- **Education**: Acts as an educational aid for entomology and environmental science.
- **Citizen Science**: Engages the public in butterfly tracking and data collection.

Model Architecture:

Model Base: `VGG16` pretrained on ImageNet

Modifications:

- Removed the top dense layers.
- Added:
  - `GlobalAveragePooling2D`
  - `Dense` with ReLU
  - `Dropout` for regularization
  - Final `Dense(75)` with softmax activation

Training Details:

- **Input Size**: 224x224 pixels
- **Loss**: `categorical_crossentropy`
- **Optimizer**: `Adam`
- **Validation Accuracy**: ~95%

Training Process:

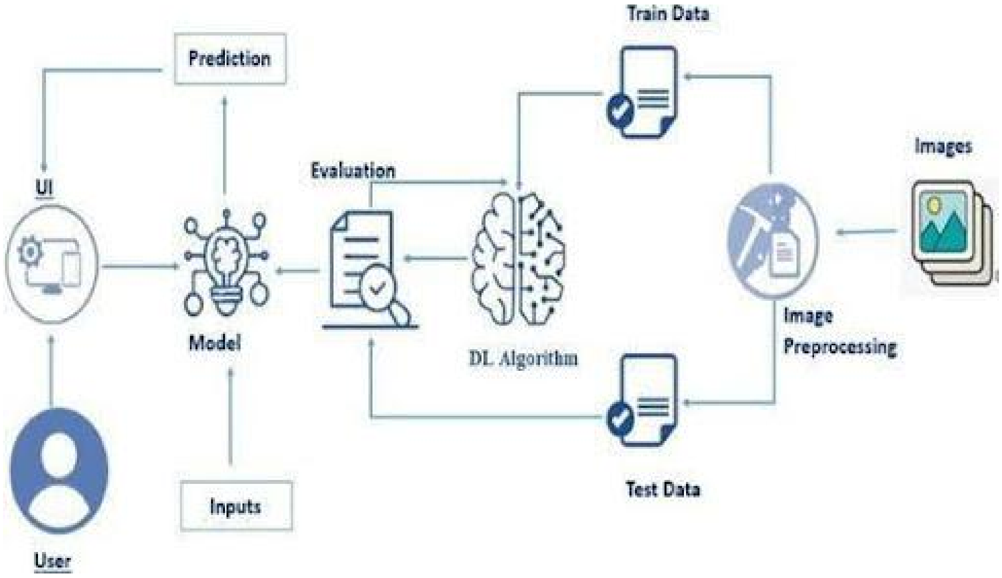
1. **Dataset Preparation**:
   - Images grouped based on `Training_set.csv`.
   - Labels encoded and images resized to 224x224.
2. **Data Augmentation**:
   - Rotation, flipping, zoom for generalization.
3. **Model Training**:
   - Used TensorFlow and Keras.
   - 80/20 training-validation split.
4. **Model Saving**:
   - Final model saved as `vgg16_model.h5`.

Dataset Details:

- **Source**: Kaggle Dataset
- **Classes**: 75 unique butterfly species
- **Train Images**: ~4500
- **Test Images**: ~500
- **CSV Files**:
  - `Training_set.csv` includes `file_name` and `category`
  - `Testing_set.csv` contains only `file_name` (for final test)

Architecture:

This architecture illustrates a complete deep learning-based image classification pipeline from user input to prediction, evaluation, and model training.



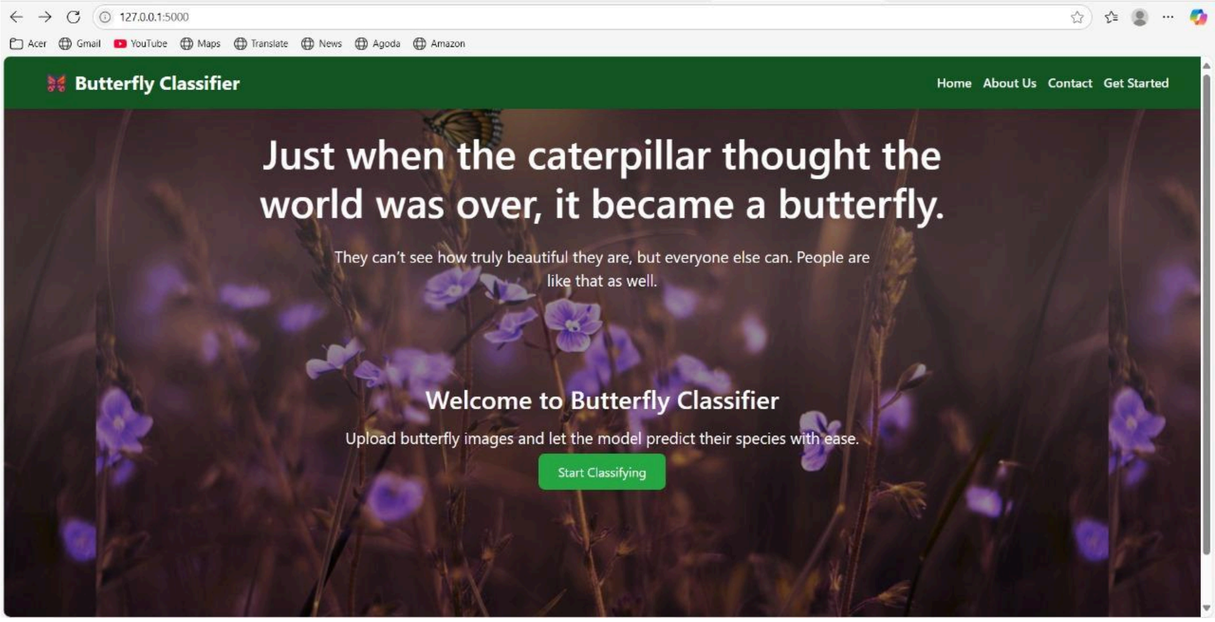
1. User Interaction (UI)

- Users interact with a visually designed web interface built using **Flask** with **HTML/CSS**.
- Through this interface, users can **upload butterfly images** for classification.
- After submission, the image is sent to the backend for processing and prediction.

2. Inputs

- The system accepts **image files** (typically `.jpg`, `.jpeg`, `.png`) as input.
- These files are securely routed to the **Flask server** and temporarily saved for processing.

Project Screenshots of Demo:

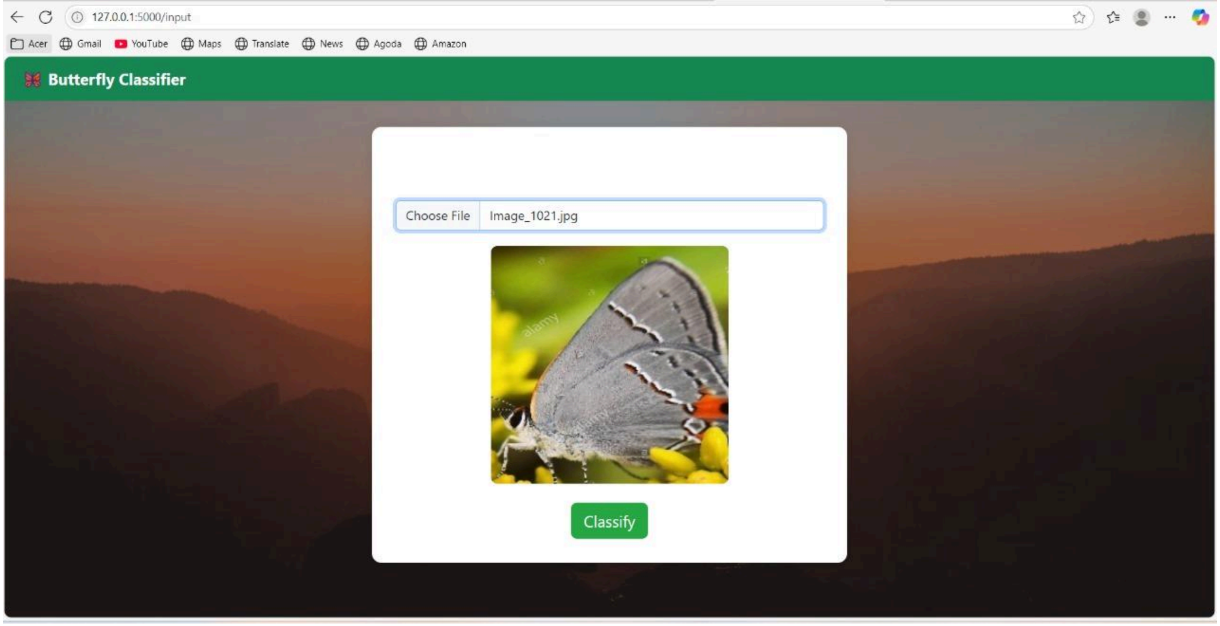


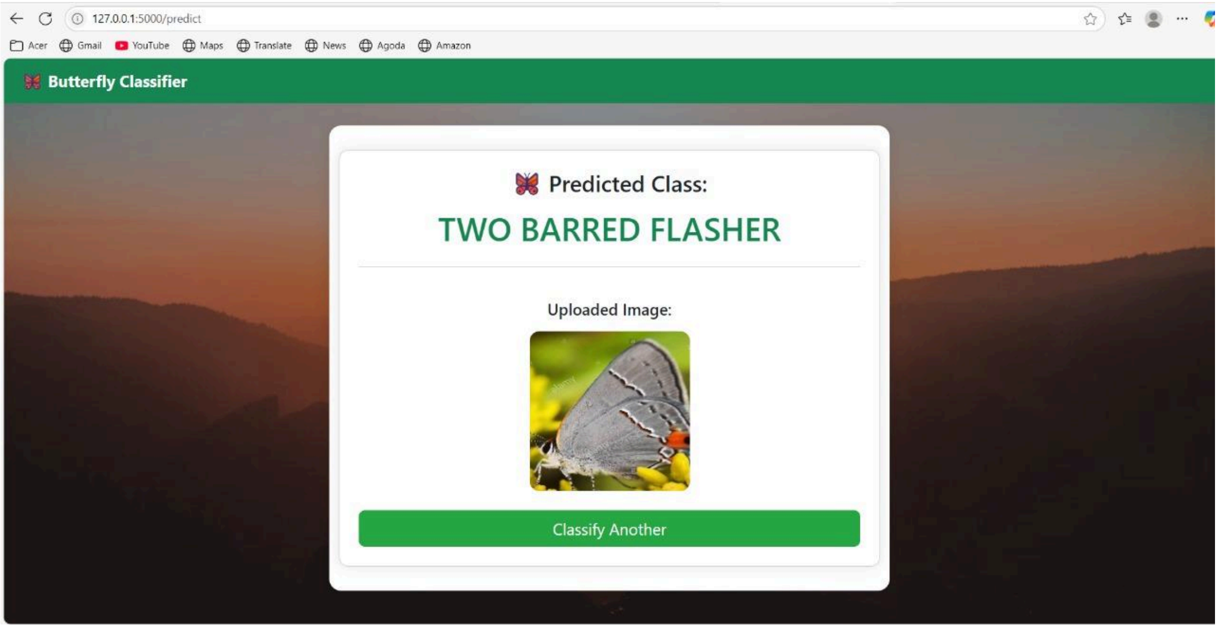
Just when the caterpillar thought the world was over, it became a butterfly.

They can't see how truly beautiful they are, but everyone else can. People are like that as well.

Welcome to Butterfly Classifier

Upload butterfly images and let the model predict their species with ease.





Predicted Class:

TWO BARRED FLASHER

Uploaded Image:

3. Model & Prediction

- Uploaded images are passed to a **pre-trained deep learning model (VGG16)**.
- The model performs **inference** and generates a predicted class index.
- This index is mapped to the corresponding **butterfly species name**.
- The result is returned to the **frontend UI** for user-friendly display.

4. Training & Test Data

- The model was trained using a labeled dataset containing **~5000 butterfly images**.
- The dataset spans **75 different butterfly classes**, ensuring wide species coverage.
- The dataset was split into **training and testing sets** to validate performance.

5. Image Preprocessing

- Every uploaded image undergoes the following preprocessing steps:
  - **Resizing** to `224x224` pixels.
  - **Normalization** (scaling pixel values to `[0, 1]`).
  - **Conversion to NumPy array** for model compatibility.
- These steps ensure consistency with the **VGG16 input requirements**.

6. Deep Learning Algorithm (VGG16)

- The system uses a **fine-tuned VGG16** model pre-trained on ImageNet and adapted to the butterfly dataset.
- Training configurations:

    - **Loss Function:** `categorical_crossentropy`
    - **Optimizer:** `Adam`
- The best-performing model was saved as `vgg16_model.h5` for future use.

7. Evaluation

- The model was thoroughly evaluated on the test dataset.
- Evaluation metrics include:
    - **Accuracy**
    - **Loss**
    - **Confusion Matrix**
- The model demonstrated **strong generalization and high accuracy** across all 75 classes.

8. Prediction Output

- After classification, the system returns:
    - The **most probable butterfly class**
    - The **butterfly name** based on class mapping
- The result is displayed back to the user within a **clean and responsive UI**, along with the uploaded image.

Frontend:

- Built using **HTML5**, **CSS3**, and **Jinja2** templates.
- Key Pages:
    - `index.html`: Introductory quote & navigation
    - `input.html`: Upload interface with preview and submit
    - `output.html`: Shows predicted butterfly species and image
    - `about.html`: Displays project goals, dataset, architecture
    - `contact.html`: Developer information and support

Backend (Flask):

- **Image Upload**: `POST /predict` saves file in `static/images`
- **Model Loading**: Loads `vgg16_model.h5` once at startup
- **Prediction Logic**:
    - Resizes & preprocesses image
    - Feeds to model
    - Retrieves prediction
    - Maps to class name
- **Response**: Rendered via `output.html` with result

📎 Requirements

Here are the libraries used:

```
Flask==2.3.2

tensorflow==2.12.0

keras==2.12.0

numpy==1.24.3

Pillow==9.5.0
```

Install using:

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

🖥️ How to Run Locally

Clone the repo:

```
git clone https://github.com/pradeepreddy/ButterflyClassifier.git

cd ButterflyClassifier
```

1.

Install dependencies:

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

2.

Run the Flask server:

```
python app.py

Open browser and go to:

http://127.0.0.1:5000
```

Conclusion

The **Butterfly Classification System** successfully integrates deep learning with a user-friendly web interface to provide accurate identification of butterfly species from images. Leveraging the powerful **VGG16 CNN architecture**, the system achieves high performance across 75 butterfly classes using a well-preprocessed and curated dataset.

The modular design ensures scalability and maintainability, while the seamless frontend-backend integration enables users to easily interact with the model in real time. This solution not only demonstrates the practical application of computer vision in biodiversity and agriculture but also provides a foundation for further advancements in species recognition.

With its clean UI, robust backend, and predictive accuracy, this project stands as a strong example of applied AI in environmental sciences and educational tools.