Report For Computer Vision Final Project

Introduction

The objective of this project is to develop a model for butterfly species classification using image analysis techniques, including image processing and machine learning. The project utilizes a dataset consisting of images of 100 butterfly or moth species. The dataset has been carefully checked for potential issues and provides training, testing, and validation subsets. In this report, we will discuss the dataset used, the methodology employed to develop the model, the results obtained, a conclusion followed by limitations for the project. The report also includes an analysis of the performance of the model and its impact on different aspects of the project.

Dataset Selection

I selected the dataset for this final project based on several key factors.

Firstly, the dataset offers a suitable and relevant domain for applying image analysis techniques. Butterflies are visually captivating creatures, and their diverse species present unique patterns and characteristics that can be captured and analyzed through image processing and machine learning algorithms. The dataset provides an opportunity to develop a model that can accurately classify different butterfly species based on their visual features.

Secondly, the dataset contains what should be sufficient images and species for comprehensive exploration and analysis. With 100 different butterfly species and a substantial number of images for each class, the dataset provides a diverse and representative sample of butterfly species, allowing for robust training and evaluation of the model.

Furthermore, the dataset's availability on Kaggle, a well-known platform for data science and machine learning, ensures easy access and reproducibility of the project. The dataset is accompanied by relevant metadata, including image labels and class information, facilitating efficient data preprocessing and model development.

Summarized, I chose this dataset for my final project because it's relevant to image analysis, has a diverse and extensive collection of data, is available on a reputable platform, and has a high-quality standard. These factors make the dataset a practical and valuable resource for developing and evaluating a butterfly species classification model.

Dataset Description

The revised dataset consists of images of 100 butterfly or moth species, each with dimensions of 224 x 224 x 3 in JPEG format. It's divided into three subsets: training, testing, and validation. The training set comprises 12,594 images organized into 100 subdirectories, with each subdirectory representing a specific species. The testing set consists of 500 images, divided into 100 subdirectories, with five test images per species. The validation set contains 500 images, organized similarly to the testing set. A CSV file named "butterflies_and_moths.csv" facilitates data organization and management, containing class id, file paths, labels, and dataset information. This CSV file simplifies data retrieval, labeling, and analysis, ensuring efficient development and evaluation of the models.

Methodology

Implementation Introduction

The approach I used to tackle the challenge was the principle of Transfer Learning. This strategy harnesses the capabilities of a pre-existing model – in this case, ResNet50, a convolutional neural network (CNN) model known for its remarkable performance in image classification tasks. Pre-trained on a vast dataset named ImageNet, ResNet50 offered a reliable foundation for our classification model.

As an initial step, I established an environment furnished with all necessary tools. This involved TensorFlow, an open-source platform for machine learning, and Keras, an interface for working with neural networks like ResNet50. In addition, libraries such as NumPy, Pandas, Matplotlib, and Seaborn were included to aid data manipulation and visualization.

Next, the process demanded the preparation of our training, validation, and testing datasets. The images were resized, pixel values normalized, and the class labels were transformed into a format suitable for subsequent stages.

In the following phase, I focused on crafting a Convolutional Neural Network (CNN) model. This task involved using ResNet50 as a base, subsequently freezing then enhancing it with layers designed specifically for our classification task. The architecture comprised a Flatten layer, a Dense layer, a Dropout layer, and, ultimately, the output layer.

Once the model was compiled, I proceeded to train it on the training dataset for a determined number of epochs which in this case was 10. Each epoch signifies a complete pass through the entire training dataset, during which the model's weights are adjusted to minimize the loss function. Concurrently, the validation dataset was employed to assess the model's performance after each epoch. This allowed me to evaluate the model's ability to generalize to unseen data effectively.

Preprocessing

The images were preprocessed before being passed into the machine learning model. They were loaded from three separate directories for training, validation, and testing. The image size was set to 224x224 pixels. These images were loaded in batches of 32.

A sample of the training images was visualized to verify the correctness of the preprocessing steps.

Model Design

The model was built using the Sequential API from Keras, with ResNet50 as the base model. The base model's layers were frozen, meaning their weights weren't updated during training. Following the base model, we added a flatten layer, a dense layer with 512 units, a dropout layer with a rate of 50%, and a final dense layer with 100 units corresponding to the number of classes in our dataset. The model was then compiled with the Adam optimizer, a learning rate of 0.0001, sparse categorical cross-entropy as the loss function, and accuracy as the metric.

This is done to take advantage of the learned weights from ResNet50, significantly reducing the training time and improving the model's performance on small datasets.

Training

The model was trained on the training dataset for ten epochs and validated using the validation dataset. The change in loss and accuracy for both the training and validation datasets were tracked and plotted over these epochs. This helped to visually assess how well the model was learning and if it was overfitting or underfitting.

Performance of the Model

The results obtained from the classification report provide valuable insights into the model's performance in classifying butterfly species. The information includes precision, recall, F1-score, and support for each class, allowing us to analyze the model's behavior on individual species.

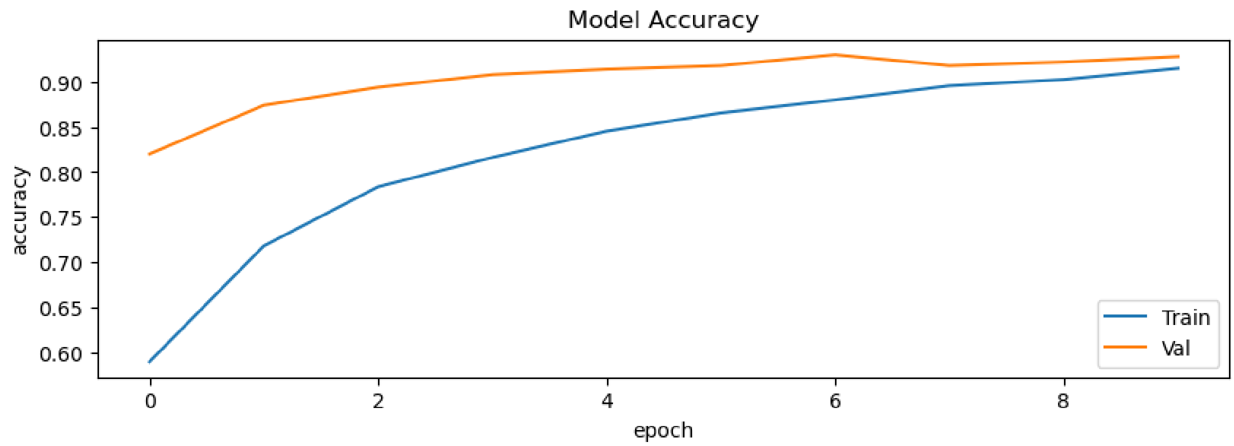
Overall, the model demonstrated high accuracy and good performance across most classes. Several classes achieved perfect precision, recall, and F1-scores, indicating that the model could correctly classify those species with high confidence. This suggests that the visual features of these species were distinctive and well-captured by the model.

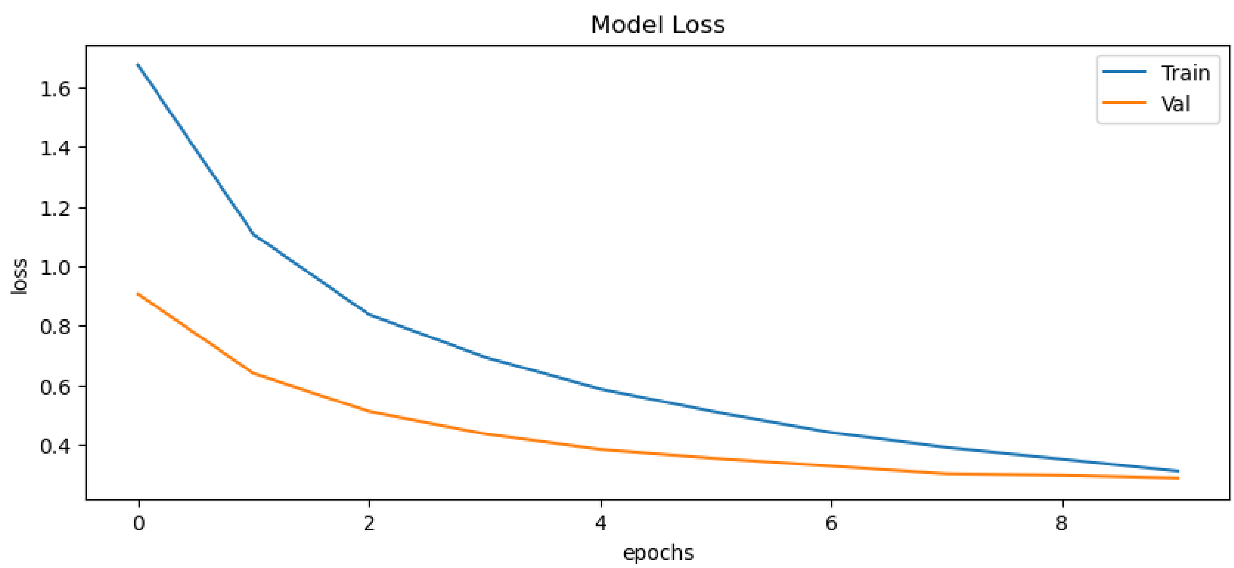
However, some classes showed lower performance metrics, such as lower precision, recall, or F1-score. It may be due to various factors. One possible reason is the similarity or visual resemblance among certain butterfly species, making them more challenging to differentiate based on the available dataset. Another reason could be the limited number of samples or imbalanced representation of certain classes, leading to less reliable estimation of their performance metrics.

The lower performance on specific classes highlights areas for improvement in the model and dataset. It signals the need for further analysis and potential adjustments to enhance the model's ability to classify those species correctly. Additional data collection, especially for classes with limited samples, can help improve the model's performance by providing more diverse examples to learn from. Fine-tuning the model architecture, adjusting hyperparameters, or exploring alternative algorithms may also be beneficial.

Model with built-in functions:

	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.94	500
Macro avg	0.95	0.94	0.93	500
Weighted avg	0.95	0.94	0.94	500





Model hits accuracy of >90% in Epoch 9/10

```
Epoch 9/10
394/394 [==============================] - 385s 977ms/step - loss: 0.3520 - accuracy: 0.9025 - val_loss: 0.2974 - val_accuracy:
0.9220
```

Conclusions

The final project on butterfly species classification was successfully accomplished with the creation of a model capable of identifying different species based on image analysis. The results revealed that the model performed well across most butterfly species, demonstrating high precision (avg. >90%), recall (avg. >90%), and F1-score (avg. >90%). However, there were some classes where the model performed less effectively, suggesting the presence of challenging classes in the dataset due to similarities among butterfly species or limited of certain species representation in the dataset.

These findings illuminate potential areas for future work. For instance, collecting more diverse and balanced data could improve the model's performance. Likewise, fine-tuning the model's hyperparameters or exploring alternative machine learning algorithms may yield better results. Also, the development of an ensemble model or the use of data augmentation techniques could potentially improve the classification accuracy of difficult classes.

Limitations

Despite the overall success of the project, there were some limitations. The main one being the potential for class imbalance in the dataset, which can negatively impact the performance of the model on under-represented classes. Also, the model's performance may be affected by the inherent similarities between some butterfly species, which makes differentiating between them more difficult.

For future work, several approaches could be considered to address these limitations. First, data augmentation techniques can be applied to increase the size and diversity of the dataset, particularly for under-represented classes. Second, alternative algorithms or deep learning architectures can be explored to improve the model's capacity to differentiate between visually similar species. For instance, utilizing Convolutional Neural Networks (CNNs) with more layers or different architectures may yield better results. Finally, ensemble learning methods, which combine the predictions of multiple models, could be applied to increase the robustness and accuracy of the species classification. The last limitation is also the lack of my experience with the field due to limited knowledge in image analysis and domain knowledge which could've been useful for choosing which preprocessing techniques to implement or if there was a better dataset to choose for this project.