Assignment 1 Instruction

Overview

The goal of this assignment is to guide you through the complete process of data preprocessing and model training, including data cleaning, dataset splitting, data loader design, and implementing a training loop for a simple neural network (e.g., CNN or MLP). You will work with a corrupted dataset containing 150 images (120 for train, 30 for test) for butterfly species classification. The dataset includes various issues, such as missing values and incorrect labels. Your tasks are to identify and mitigate the effects of these corruptions, split the data using an appropriate train–test ratio, and implement a basic training loop using PyTorch and relevant Python libraries (e.g., NumPy, Pandas). All work must be done in a **Jupyter Notebook**, with clear comments explaining your code and demonstrating your understanding to critical concepts. Results must be clearly presented, and your code should be bug-free and reproducible on any machine with the same software setup

Resources

- **A folder of 150 images** in .jpg format
- **A CSV annotation file** containing image filenames, labels
- **Jupyter Notebook Template.** All works must be displayed clearly in Jupyter Notebook for submission. No additional files are allowed as supplementary materials.

Task 1: Data Preparation (8 %)

All important steps must be explained with one to two lines comments to show your understanding of the implementation. For example, explain why we need data loader, or what data corruption have been identified.

- **Data Clean (2%):** Identify and remove any rows in the annotation file that do not conform to the provided data description. This includes, for example, label values outside the valid range (e.g., labels should be 0-10, but 11 appears in the data), missing or corrupted image files, and other inconsistencies. Produce a cleaned version of the annotation file. You are expected to implement a data cleaning program **(NOT hand cleaning)** that clean your data by **removing those corrupted data**. Hint: print your label distribution / visualization / missing value check may be helpful to spot the corruption.
- **Image Processing (2%):** Resize all images to a fixed target size (e.g., 64x64 pixels) and **normalize** the pixel values to the range [0, 1].
- **Data Splitting and Loader Implementation (2%):** Split the cleaned dataset into training, validation, testing by a proper ratio. Implement separate data loaders for all sets, ensuring that there is no overlap between all sets.

- **Data Augmentation (2%):** Implement **at least two** basic data augmentation techniques (e.g., horizontal flip, rotation) in the training data loader to increase dataset diversity during model training.

Task 2: Training Loop (7%)

- **Model Architecture (2%):** Implement a neural network for classification using a simple CNN as specified. All models must be implemented using the basic network block from Pytorch (e.g. nn.Linear()) using OOP. Any pre-defined or pre-trained model is not acceptable.
- **Model Training and Evaluation (5%):** Develop a training loop that updates the model parameters using an optimizer. The training loop should iterate over the training data in batches, compute loss and accuracy, and print the results for each epoch. You need to initialize 3 different parameters setting for you model and select the best one for testing. A brief discussion (up to 500 words) on the choice of parameter set, components in model and critical training step. For example, why SGD/Adam as optimizer is better and why gradient cleaning.

Useful Resources

Overleaf Quick Tutorial: https://www.youtube.com/watch?v=xcTN4F3l9Ds

PyTorch Quick Tutorial: https://www.youtube.com/watch?v=V_xro1bcAuA&t=23647s

TensorFlow Quick Tutorial: https://www.youtube.com/watch?v=tPYj3fFJGjk&t=6685s

Necessary Environment

Python, Numpy, Pandas, Pytorch, OpenCV