Task

In this technical task, the applicant will build and evaluate a small Image Retrieval setup. Go through the nice presentation from Kevin McGuinness on Content Based Image Retrieval with Deep Learning to gain a better understanding of the Image Retrieval problem and modern approaches on solving it.

Task 1: Create a GitHub Repository

Create a repository, with a name of your choice, under your GitHub account. The purpose of this repository is to commit all the sources developed and information gathered to solve the Tasks 2, 3 and 4.

Hints and Tips:

- It is highly recommended to add a **README.md** with nice documentation to allow someone to run and reproduce your solutions for Tasks 2, 3 and 4.

Requirements:

- Share the link of this repository once all the Tasks are completed.

Task 2: Prepare Data

The Caltech-UCSD Birds-200-2011 (CUB200) Dataset contains 11788 images of 200 different Bird Species. It is one of the commonly used datasets in the research community to benchmark novel CBIR algorithms. Each image in the CUB200 dataset is annotated with class label, bounding box, part locations and binary attributes. Solving Task 2 does not require the part locations and binary attributes, so ignore these annotations and consider only the class label and bounding box annotations. Go through the dataset's official website for more details.

Follow the below steps to complete Task 2:

1. In the official website, follow the **Download** section to grab a copy of this dataset. Download only the following:
    - All Images and Annotations (this zip file contains images and annotations)
    - README (this text file describes the contents of above zip file in detail)
2. Run an Exploratory Data Analysis (EDA) on the downloaded dataset to get a better overview on the and skim through a few samples.
3. Now, split the data into training and testing subsets. Follow the data splitting strategy described in **Section 7.1** of the paper **Deep Metric Learning via Lifted Structured Feature Embedding.**

Hints and Tips:

- Running EDA in a Jupyter or Colab Notebook is recommended.
- Pandas and Matplotlib are some of the tools used extensively in EDA.
- Either CSV or JSON file formats can be used to save the training and testing subsets information.

Requirements:

- Push the scripts or notebooks related to EDA to the GitHub repository created in Task 1. If a Colab notebook has been used, provide its link in the README file.

Task 3: Extract Embeddings

Convolutional Neural Networks (CNNs) are a category of Artificial Neural Networks used widely in Deep Learning. They work exceptionally well on problems involving image as an input. Some of the applications where CNNs are commonly used are Image Classification, Object Detection, Image Retrieval, Semantic Segmentation, etc.

Normally, CNNs are pre-trained on large datasets and are later adapted for downstream tasks. For instance, a ResNet model could be pre-trained on ImageNet dataset to predict 1000 classes. Once pre-trained, this ResNet model can be used as a base network, just by removing its last Fully-Connected (FC) Layer, for training an Image Classifier on the CUB200 dataset. Similarly, such pre-trained models can also be used for Transfer Learning and Feature Extraction.

For this task, the focus will be set on Feature Extraction from pre-trained CNNs. Feature Extraction or Extracting Embeddings from a CNN is a technique of encoding an image into a vector representation which captures the informative content in the image. Often the terms Feature, Embedding or Vector Representation are used interchangeably. The features from pre-trained CNNs are normally generic enough to be used for Image Retrieval tasks.

Now follow the below steps to extract embeddings from the train and test data prepared in Task 2.

1. Pick a pre-trained CNN which is readily available from PyTorch or TensorFlow.
2. Extract train and test data features from the chosen pre-trained model.
3. Save the extracted features as a numpy file which will be required solving Task4.

Hints and Tips:

- PyTorch Image Models provides a simple api to extract features from pre-trained models.

Requirements:

- Push the scripts or notebooks related to feature extraction to the GitHub repository created in Task 1. If a Colab notebook has been used, provide its link in the README file.

Task 4: Evaluate

Now the image features from both training and test data are available, next step is to evaluate how good the chosen pre-trained model is solving CUB200 Image Retrieval task.

Steps for solving Task 4:

1. List a set of metrics that can be used to evaluate Image Retrieval models.
2. Describe what information each of the listed metrics will convey.
3. Describe how the training and test features (extracted in Task 3) will be used to compute evaluation metrics.
4. Finally, compute the listed evaluation metrics.

Hints and Tips:

- The same evaluation metrics used in Information Retrieval are also used for Image Retrieval.
- Leaderboard of Image Retrieval on CUB200

Requirements:

- Solutions for steps 1, 2 and 3 should be documented in the README file of the repository.
- Push the scripts or notebooks related to computing evaluation metrics to the GitHub repository created in Task 1. If a Colab notebook has been used, provide its link in the README file.