**Image Clustering Group Assignment**

Total Marks 100

**Scenario:**

*Your company collects images from the social media pages of the client’s competitors and saves the images into a common folder. As a Data Scientist, your task/objective is to take all collected images in the common folder as inputs and group (cluster) visually similar images together.*

However, in this assignment, we shall use a public dataset. However, your code should be flexible enough to accommodate any unknown dataset [e.g., the dataset explained in an earlier paragraph].

Dataset: The [Flower dataset](https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz) has 3669 images of five types (daisy, dandelions, roses, sunflowers, tulips) of flowers.

|  |  |
| --- | --- |
| Flower name | Total images |
| Daisy | 633 |
| Dandelion | 897 |
| Rose | 641 |
| Sunflower | 698 |
| Tulip | 799 |

**Dataset Link:** [flowers.zip](https://saskpolytech-my.sharepoint.com/:u:/g/personal/muktadira_saskpolytech_ca/EYpkhq3BcU1IoG1Mq0gfD6wB7KIaIW7C6ZFighLdnxtQCg?e=W3jval)

* You can decide the number of clusters/groups for an unknown dataset, and there is no right/wrong answer to how many clusters there should be.
* The important part is that the images in the groups should be visually similar. For example, for the Flower dataset, there should be five clusters. However, you should explore with more and fewer clusters.
* One thing you will need to do is to convert each image to some numerical representation. For example, each image could be converted into a feature vector/embedding. You may have some ideas of how to do this. If not, then here is one idea which uses pre-trained ImageNet embeddings:

I have extracted all images belonging to into a single folder

```python

from tensorflow.keras.applications.resnet50 import ResNet50

from tensorflow.keras.preprocessing import image

from tensorflow.keras.applications.resnet50 import preprocess\_input

import numpy as np

import tensorflow as tf

model = ResNet50(weights='imagenet', include\_top=False, pooling='avg')

img\_path = '../your\_path/imagge01.jpg'

img = image.load\_img(img\_path, target\_size=(224, 224))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0) x = tf.cast(x, tf.float32)

x = preprocess\_input(x) features = model.predict(x)

```

The above code snippet was taken from here: <https://keras.io/api/applications>. You can read about different types of pre-trained feature extractors there. It's important that the images are resized and pre-processed correctly for the particular model chosen. That is fine if you want to use a different library, such as Pytorch, and/or a different approach.

**The above code example is simply an idea. Feel free to use it or something else.**

* Once you have clustered the images, your code should output a dictionary, which is keyed by `cluster\_id`, and the values should be the image IDs (image file name) that belong to that cluster. For example, if you found 2 clusters the dictionary might look like this:

```python

cluster\_mapping = {0: [image01, 'image06'],

1: ['image02', 'image10']}

```

* **Physically move similar images into separate folders. You should try your clustering performance with another dataset: for example, you can collect N cats and dogs images into one folder, and then run the clustering and copying to two different folders, and then count how many images have been put together correctly. N >=20**
* Evaluate your method of clustering the images - using evaluation methods/metrics of your choice. Your code should output the evaluation results. **You should evaluate K-Means, and any two other clustering methods from Scikit-Learn library (** [**https://scikit-learn.org/1.5/modules/clustering.html**](https://scikit-learn.org/1.5/modules/clustering.html) **) and compare their performance.**

Use a Jupyter Notebook to combine a runnable code with text description and visuals. In the writeup include a short (few sentences) explanation for each of the following:

● Explanation of your choice of the numerical representation of images,

● Explanation of your choice of the clustering method,

● Explanation of your choice of the clustering evaluation method, as well as:

● A visual representation of clusters that shows up to 10 random images for up to N (N should be less than the number of clusters created) biggest clusters.

**Include a readme file that specifies how to run your code (e.g., if a library is needed, how to install it, where to put the input images, etc.). I want to run your code.**

**Note: Your Jupyter notebook should run without any issues.**

|  |  |
| --- | --- |
| Criteria | Assignment Marks |
| Image Preprocessing and extracting numerical representation of each input image | 30 |
| Image clustering and clustering performance evaluation | 35 |
| Output dictionary and images are sorted into related folders, with flower and cat-dog images | 15 |
| Showing random images from each cluster of either dataset | 10 |
| Writeup | 10 |
| Total | 100 |

Thank you!