CS452/552 – Data Science with Python
Assignment-2

Image Compression by K-Means Clustering
Due: 13.12.2020 – 23.55

created by Furkan Kınlı, for questions → e-mail: furkan.kinli@{ozu, ozyegin}.edu.tr

1. Definition

In this assignment, you will implement a **Jupyter notebook** that solves the problem of image compression by using one of the well-known clustering methods, which is called K-Means Clustering. We will **NOT** provide a baseline notebook for this problem, so you need to implement the notebook solving the image compression problem from scratch. We expect you to pre-process the images that we provide as we expect, fit K-Means clustering model that finds C number of cluster centers just by looking the pixel values, assigns the pixel values of the closest cluster center to each pixel, and lastly report the performance of your clustering model **qualitatively** and **quantitatively**. The data contains 5 unique images with different sizes, you should apply all methodology to solve the problem on these 5 images. Note that you do not need to split the data.

In this assignment, you are only allowed to use the scientific computation libraries, which are introduced in the lectures (NumPy [docs], Pandas [docs], Scikit-learn [docs]), and also you are free to use any Image Processing library (e.g. OpenCV [docs], Pillow [docs]). However, we kindly expect you **to conduct a comprehensive experimental setup and to visualize the data and the results in your notebooks**, where each cell introduces one thing and has a comment for what it does. The details of the task are in the following sections.

In the report, you are expected to introduce the project and the aim of this project, to explain the algorithms/models employed in this project in detail. Also, it is expected to show the experimental setup (i.e., each parameter in your model or each image processing method that you applied), and to report the overall performance in given metrics for your clustering model. Moreover, descriptive tables, plots and figures are required **both to observe the images better and to show the results for the settings**. Please do **NOT** forget to add visualization for the images and the compressed results in your notebooks.

We strongly recommend & ask you to read about the algorithms and the problem before starting to implement your assignment.

2. Implementation Details

For this assignment, we will **NOT** provide a baseline Jupyter notebook, which means you will implement the notebook for image compression by K-Means clustering from scratch. You may not use more than one notebook, which means **you need to have a single "*.ipynb" file that is responsible for all tasks**. For each cell, a descriptive comment in the first line is **must**. Please use **the newest** version of the libraries.

The details of this task as follows:

- Giving information about K-Means Clustering in your report. (5 pts)
- Reading the images, pre-processing (i.e., resizing to 256x256x3, reshaping them into 2-d array) and visualizing them in notebook. Do not forget to add visuals of the images to your report. (5 pts)
- Finding how many bytes a single image is consist of, and report the numbers uniquely. (2.5 pts)
- Finding how many unique colors a single image contains, and report the numbers uniquely. (2.5 pts)

For each image given:

- Initialize your K-Means clustering models (2.5 pts)
- **Fit the image in form of 2-d array (size: Nx3) where N equals to "height x width" in order to find K cluster centers of the pixel values of the image where K is the positive power of 2, up to 256 (2^8). (10 pts)**
- For each setup with different number of clusters, re-assign each pixel value as the closest cluster center to it. (10 pts)
- Find the image size (how many bytes) of the compressed images constructed by each number of clusters (8 different compressed images). (5 pts)
- Visualize these 8 different compressed images in the notebook. (5 pts)
- **Calculate the following metrics without using any library (from scratch, pure python or Numpy) for each number of clusters. (20 pts)**
    - **WCSS: Within Cluster Sum of Squares**
    - **BCSS: Between Cluster Sum of Squares**
    - **Explained Variance (Silhouette Coefficients)**
- Create a table in your report which shows the name of images, the number of clusters, the name of centroid colors (see webcolors), WCSS, BCSS, Explained variance and the size of compressed image for each number of clusters. Note that you may use Pandas DataFrame to generate this kind of table in the notebook, and are free to use any method to generate the table in the report. (10 pts)
- **Calculate the optimal elbow without using any library (from scratch, pure python or Numpy), and discuss them in the report, which number of clusters should we choose to obtain the compressed image with the best possible quality, but least memory requirement. (consider the trade-off between explained variance and image size) (20 pts)**
- Compare the explained variance and image size of the original images and the best quality compressed image. (2.5 pts)
- Note that the points given in this section will be calculated for 5 images in total. For example, initializing the model gives 0.5 pts for a single image, and if it would be completed for all 5 images, then 2.5 pts.

3. Criterion

Notebook & Report: 100 Points

K-Means Clustering and Image Compression problem are well-studied in the literature and on online resources, and thus there are many different coding examples online. We have almost all of them, at least we have a chance to compare your notebooks with our collected online resource database for this problem. Please do **NOT** try to use them directly. In any circumstances of copying online resources directly, **you will get 0 points**.

Please do **NOT** include any other 3rd-party library to your notebook (except NumPy, Pandas, Scikit-learn, OpenCV, Pillow, webcolors), because you do not need them. Including such libraries may cause a problem for running your code in different local environments (e.g. dependencies). For any dependency on run-time that **TA can solve**, **you will get -15 points penalty**; for any dependency on run-time that **TA cannot solve, you will get -30 points penalty**.

You need to write a detailed report to explain your design and solution**. 1 PARAGRAPH CODE EXPLANATION IS NOT A REPORT, and such submissions will be ignored, and you will get 0 points, even if your notebook does something**. Please follow the technical report writing rules (i.e. *Introduction, Methodology, Implementation Details, Results, Conclusion*). It is **must** to add the visual contents that you extract in your notebooks to your reports.

The notebook file (*.IPYNB) and the report file **(*.PDF**) should be zipped (**.ZIP**) together. The filename of final submission file should be in the format of "**NAME_SURNAME_ID_hw2.zip**". Please follow this structure for your submission. You should **NOT** include the images (.JPEG) which is provided by us to your submission. Not following this structure will be resulted as **"-10 points penalty for each" (file extension, filename, zipping, not including the data) without any excuse**.

Late submission is allowed for **1 extra day with -20 points penalty**.

You may discuss the algorithms and so forth with your friends, but this is an individual work. Therefore, you have to submit your original work. **In any circumstances of plagiarism, first, you will fail the course, then the necessary actions will be taken immediately.**