Applied Mathematics and Statistics

Project Report
Color Compressio

Lê Quốc Trung - 20127369

I. Project description

1. Idea

- One image may contain many pixels. And one pixel contains 3 numbers R, G, B, which are in range [0, 255]. These 3 numbers would make up a color. So we have 2563 colors, which make the file heavy.
- This project is for the purpose of reducing the number of colors on the image down to k_cluster.
- We first pick k pixels in the image as centroids and then separate others to these clusters. And every loop, we update the new centroids and separate again. This process repeats until the centroids do not change or maximum iterators are reached.
- We finally have a new image with less colors.

2. Libraries and modules needed for project

- Numpy
- Module pyplot from matplotlib
- Module Image from PIL

3. Enviroment

- Jupyter Notebook

II. Algorithms

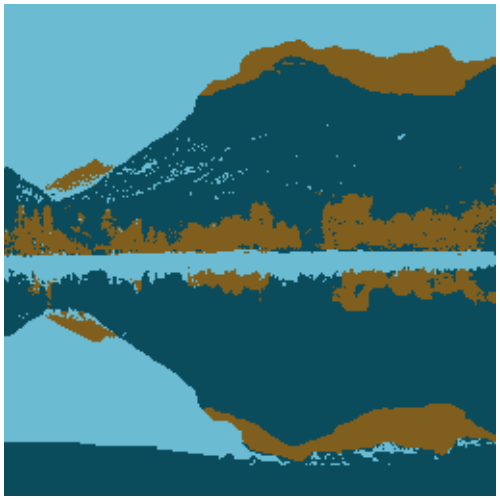
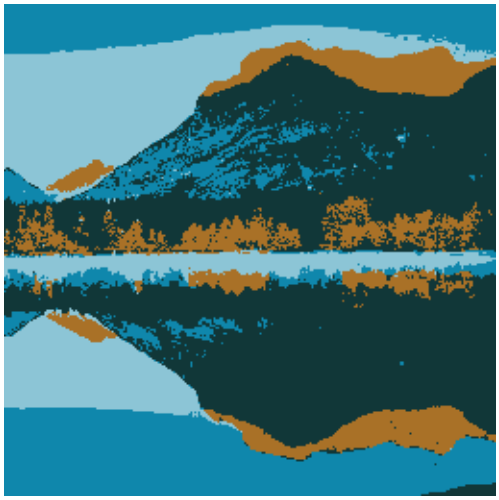
Step 1: Use PIL to read image
Step 2: Reshape original image from 3 dimension to 2 dimension
Step 3: Kmean

- Random init K centroid:
  - 'random' --> centroid has `c` channels, with `c` is initial random in [0,255]
  - 'in_pixels' --> centroid is a random pixels of original image
- Update centroids:
  - Relabel each pixel with label of nearest centroid
  - Find new value for centroid by calculate with mean of all pixel in that centroid (use np.linalg.norm)
  - Until reach max_iteration

III. Result

Original image:

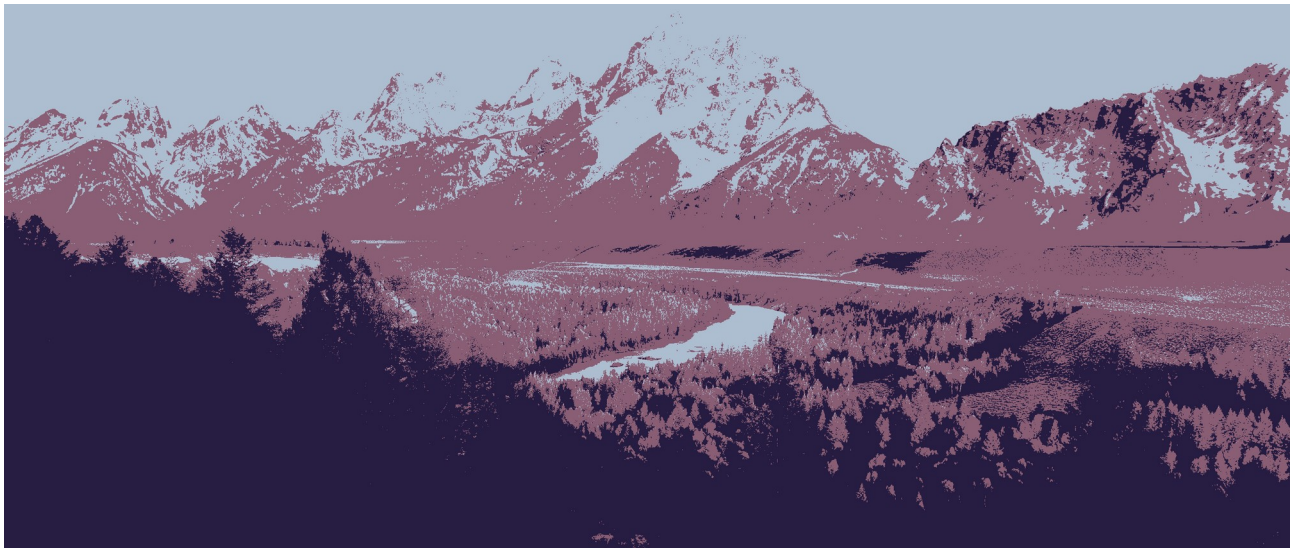
Result (in 3, 5, 7, 9 centroids):

Try with higher resolution image (2506x1080):

Original image:

Result (3, 5, 7 centroids):



IV. Comment

- These picture are landscape, it natural usually have more color than other type of image
- In 3 centroids, we can only see the ‘shape’ of the picture, just near to black-white picture, hard to imagine how the real image look like
- When have more centroids (5, 7) or even 9 centroids, it very easy to guest how original image look like but also can save lot of memory to store the image. The higher k_cluster is the more beautiful the image is
- Though being compressed, we still see the structure of the origin image.

V. Reference

https://www.geeksforgeeks.org/python-pil-image-open-method/

https://ai538393399.wordpress.com/2020/09/29/k-means-clustering-algorithm-without-libraries

https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670