UNIVERSITY OF SCIENCE

VIETNAM NATIONAL UNIVERSITY

IMAGE COMPRESSION

Lecturer

Vu Quoc Hoang
Nguyen Van Quang Huy
Le Thanh Tung
Phan Thi Phuong Uyen

Ho Chi Minh City, June 11th, 2022

TABLE OF CONTENTS

IMAGE COMPRESSION

Applied Mathematics And Statistics

1. INFORMATION

Project name: Image Compression

Lecterer:

- Vu Quoc Hoang
- Nguyen Van Quang Huy
- Le Thanh Tung
- Phan Thi Phuong Uyen

Class: 20CLC08

Student:

- **ID:** 20127039
- **Name:** Tran Dam Gia Huy

2. CONTENTS

2.1.Approach and Implementation

2.1.1. Library Support

```
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
```

2.1.2. Image Handle

```
def ReadImage(imgName):
    imgOrg=Image.open(imgName)
    return imgOrg

def ImageToArray(imgName):
    imgArr=np.asarray(imgName)
    imgRes = np.reshape(imgArr, (imgArr.shape[0] * imgArr.shape[1], imgArr.shape[2]))
    return imgArr, imgRes

def RecoverShape(img2d, img3d):
     img2d = np.reshape(img2d, (img3d.shape[0], img3d.shape[1], img3d.shape[2]))
     return img2d
```

- Use the Image in PIL Library
- **ReadImage**(imgName): Read a image by PIL library on device by passing imgName as parameter which inputted by user
- **ImageToArray**(imgName): Convert an imgName to 3D array by asarray in numpy and reshape it into 2d to easy implement
- **RecoverShape**(img2d, img3d): After execute the algorithm to compress image, recover the 2d array to orginal 3d array

2.1.3. Input Handle

```
def Input():

    # Input image name
    imgName = input('Enter image name: ')

    # Input image format
    while(True):
        imgFormat = input('Enter output format ( png or jpg ): ')
        if(imgFormat=='png' or imgFormat=='jpg'): break

    #Input init_centroid
    while(True):
        init_centroid = input('Enter Centroids Initialization ( in_pixels or random ): ')
        if(init_centroid=='in_pixels' or init_centroid=='random'): break

    #Input the number of clusters
    k_clusters = int(input('Enter the number of clusters: '))
    return imgName, imgFormat, init_centroid, k_clusters
```

User inputs some variables as imgName, imgFormat, init_centroid, k_clusters by input()

2.1.4. Kmeans Algorithm

Initialize Variables

```
def kmeans(img_1d, k_clusters, max_iter, init_centroids='random'):

    # INITIALIZE VARIABLES
    centroids = []
    sample_Random_Array=[]
    labels = [ [] for i in range(k_clusters)]
```

- cetroids is an array to store k_clusters color, each element is a [R, G, B] value

- sample_Random_Array is an array to random the index of image array. I will explain more in the next step
- labels is an 2d array to store clusters on which each pixel belongs to. Example, pixel 1 and 2 belongs to cluster 1, pixel 3 belongs to cluster 2, then labels is [[1,2],[3]]

Initialize Centroids

Because of two different modes in initialize centroids (in_pixels and random), so I have 2 cases that user inputs to parameter init_centroids:

```
    # INITIALIZE CENTROIDS
        # Random in pixels
    if(init_centroids=='in_pixels'):
        while(len(sample_Random_Array)!=k_clusters):
            while(True):
                temp=np.random.randint(img_1d.shape[0])
                if(temp not in sample_Random_Array):
                    sample_Random_Array.append(temp)
                    break
        for i in sample_Random_Array:
            centroids.append(img_1d[i])

        # Free random in [0,255]
    if(init_centroids=='random'):
        while(len(centroids)!=k_clusters):
            while(True):
                temp=[np.random.randint(255) for i in range(3)]
                if(temp not in centroids):
                    centroids.append(temp)
                    break
```

In_pixels mode

- Random a number index from 0 to the number of pixels and save it to temp
- Store each random number index in an array sample_Random_Array without duplication
- Append the color to centroids (color is one of the elements in image whose index is an random number is sample_Random_Array)

Random mode

- Random 3 properties RGB values and store it into array temp
- If array temp does not exist in centroids, the centroids will append temp

Update Centroids

After initializing centroids, we need to group samples into clusters and update centroids.This step will in loop with max_init times, so here I define max_init=10 to save time in compiling

```
# UPDATE CLUSTERS AND CENTROIDS
for i in range(max_iter):
    # clusters
    labels = [ [] for i in range(k_clusters)]
    for index, sample in enumerate(img_1d):
        minDis=10000; indexCluster=0
        for j in range(len(centroids)):
            if( np.linalg.norm(sample-centroids[j]) < minDis ):
                minDis = np.linalg.norm(sample-centroids[j])
                indexCluster = j
        labels[indexCluster].append(index)

    # centroids
    centroids=np.array([[0 for _ in range(img_1d.shape[1])] for _ in range(k_clusters)])
    for index, clus in enumerate(labels):
        if(len(clus)==0): continue
        centroids[index]=np.mean(img_1d[clus], axis=0)

return centroids, labels
```

Update clusters:

- Initialize 2d array labels to story clusters's index on which pixel belongs
- Use enumerate to collect indexes and colors of image array
- Calculate the distance of each color to centroids's colors by norm in numpy.linalg
- Store the index of centroids that has minimum distance in indexCluster
- Store the index of pixel's color into corresponding clusters in labels (the clusters have minimum distance to pixel's color, here is indexCluster)

Update centroids

- Initialize 2d centroids with 0
- Use enumerate(labels) in loop to access index and data of each clusters in labels
- Use np.mean to calculate the average distance of all colors in the cluster
- The index'th value of centroids is equal to the average distance of corresponding cluster
- This step that update labels and centroids will be executed 10 times before return centroids and labels

2.1.5. Change color

After executing Kmeans Algorithms, each pixel is grouped into a cluster which includes the centroids. So now, I change all pixels's color are equal to centroids'color in the conrresponding clusters. Finally the number of colors in image is similar to the number of clusters in label.

```
def ChangeColor(labels, centroids, img2d):
    for i in range(len(labels)):
        if(len(labels[i])!=0):
            for j in labels[i]:
                img2d[j]=centroids[i]
    return img2d
```

2.1.6. Main function

The main function to execute all of the code. After changing the color, I will save the image by **<filename>_out.<imgFormat>** which imgFormat is inputted by user, <filename> is the same with file name of the orginal image.

2.2.Results

Original Image

Image Compression (in_pixels centroids)		
K=3	**K=5**	**K=7**
		
K=11	**K=20**	**K=50**
		

Image Compression (random centroids)		
K=3	**K=5**	**K=7**
		
K=11	**K=20**	**K=50**
		

2.3.Evaluation

There is some evalutions after executing the algorithm:

- The origin image is 200x300 with 41107 different colors
- After compressing, the number of colors in the images is reduced with the specifical k colors but the structure of the image is still not change.
- When increasing k values, in both methods to intialize centroids have a change in colors in different parts of image (Example: in_pixes centroids,k=5 and k=3, the color of the sky and water is changed. Neveratheless, the other, it has a shift in bus' colors and shadow' colors)
- The color in in_pixels centroids initialization method is closer to the orginal image than random method
- In last test case (k=50), in_pixels centroids initialization method, it is an detail abundant in the sky's color, while the other is in the shadow's color

Overall, there is a correctness and success in the algorithm. It returns results as I expects.

3. REFERENCES

Kmeans Algorithm

- https://www.youtube.com/watch?v=vtuH4VRq1AU

Matplotlib library

- https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html.

Numpy library

- https://numpy.org/doc/stable/reference/generated/numpy.asarray.html
- https://numpy.org/doc/stable/reference/generated/numpy.reshape.html

Pillow library

- https://pillow.readthedocs.io/en/stable/reference/Image.html

Save image without frames

- https://stackoverflow.com/questions/8218608/scipy-savefig-without-frames-axes-only-content

Show image without axis in plt

- https://stackoverflow.com/questions/9295026/matplotlib-plots-removing-axis-legends-and-white-spaces

Link image

- https://unsplash.com/photos/6mze64HRU2Q