

Assignment Project Report K-Means Clustering: Image Segmentation

NAME: RISHI RAJ SINGH

AI AND ML B-4

Problem Statement :

Take a bright colorful image (Eg: image having fruits in it) and implement image segmentation using K-Means. You can first try to implement K-Means on iris dataset to understand its working and then extend the same logic, using the image pixels as the data points. Hint: All the K centroids will represent a color and therefore, you can initialize all the pixels to belong to a cluster randomly and then start the training of the centroids.

Prerequisites:

- Software: Python 3

Tools:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- OpenCv

Method Used;

Image segmentation is the process of partitioning a digital image into multiple distinct regions containing each pixel with similar attributes i.e. classification of an image into different groups. There are different

methods, and one of the most popular methods is the k-means clustering algorithm. K-Means clustering algorithm is an unsupervised algorithm, and it is used to segment the interest area from the background.

Implementation:

1. Load all required libraries

```
In [8]: import matplotlib.pyplot as plt  
import numpy as np  
import matplotlib.image as mpimg
```

```
In [9]: def Read_image(img_file):  
    img=mpimg.imread(img_file)  
    plt.imshow(img)  
    return img
```

2. Loading Data

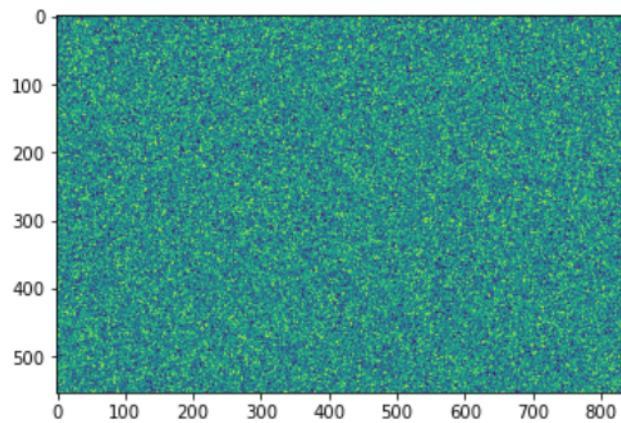
```
In [10]: img=Read_image("./fruits.jpg")
```



2. Visualizing data libraries

```
In [14]: k=2
label_arr=np.zeros(img.shape[:2])
for i in range (img.shape[0]): #through each rows
    for j in range(img.shape[1]):#through each columns
        label_arr[i,j]=np.random.choice(k)
label_arr=label_arr.astype("int32")

print(label_arr)
plt.imshow(label_arr)
```



3. Image Segmentation using K-means

```
In [12]: ➤ def Kmeans(img,label_arr,k,maxIter):  
    mean_old=init_mean(k,img,label_arr)  
    for iter in range(maxIter):  
        new_label=label_update(mean_old,img,label_arr)  
        mean_new=mean_from_label(k,mean_old,img,new_label)  
        print("the mean obtained at{i} iteration is {mean}".format(i=it,mean=mean_new))  
        label_arr=new_label  
        mean_old=mean_new  
    return mean_new,label_arr
```

```
In [13]: ➤ def segment_image(img,label_arr,mean_cent):  
    seg_img=np.zeros(img.shape[:2])  
    for i in range(img.shape[0]):  
        for j in range(img.shape[1]):  
            k=label_arr[i,j]  
            seg_img[i,j]=mean_cent[k]  
    segt_img=seg_img.astype("uint8")  
    plt.imshow(segt_img)
```

```
In [*]: ➤ mean_cent,label_arr=Kmeans(img,label_arr,k,5)
```