**MACHINE LEARNING ASSIGNMENT-2**

Name- Shubham Bhawsar

SR- 21467

**Introduction**

Clustering is a popular technique in machine learning and data mining used to group similar data points together. K-means clustering is a widely used and effective clustering algorithm that partitions a given dataset into K clusters based on the similarity of data points. In K-means clustering, the K refers to the number of clusters that we want to create.

The K-means algorithm works by dividing the data points into K clusters in such a way that the sum of the distances between the data points and the mean point of their assigned cluster is minimized.

**Dataset**

Image of size (512, 512, 3) pixel



**Flow of Code:**

**Step1**. First we take image as input in main.py file and the

**Step2**. Then we apply K-means on the image for different values of K that is [2,5,10,20,50]

**Step3.** Now for each K we fit the image by calling fit() function in model.py file from where real K means clustering code starts

***Fit() function:***

The first part of the function initializes the cluster centers by randomly selecting K data points from the dataset. The function then enters a loop where it assigns each data point to the closest cluster based on the Euclidean distance between the data point and the cluster centers. This is done using the np.argmin() function which returns the index of the smallest distance.

The function then updates the cluster centers by computing the mean of all data points in each cluster. If the new cluster centers are close enough to the previous ones (as determined by the epsilon parameter), the algorithm stops. Otherwise, the process continues until the maximum number of iterations is reached.

**Step4.** Now we call replace\_with\_cluster function which replaces the pixel value by its cluster mean pixel

**Step5.** We calculate the mean square error between original image and image obtained after replacing pixels that is image obtained from step 4

***Replace\_with\_cluster\_centre() function:***

This function takes Image as input and we have the information of each data point that which data point belong to which cluster stored in

*self.close* array and with help of this array we change each pixel of original image by replace it with that clusters mean.

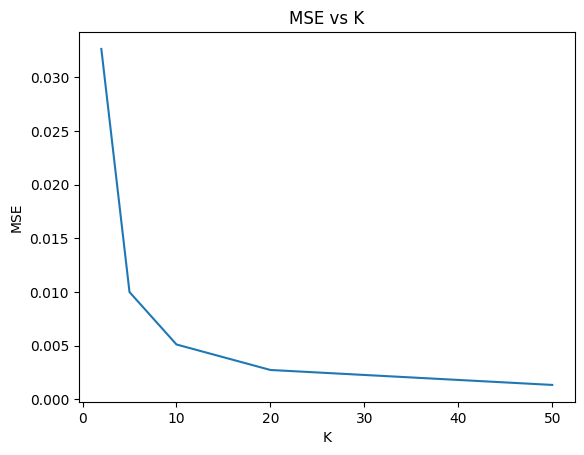
***error() function****:*

It take original image and changed image array as input and calculate the mean square error between the values of array that is pixels for image.

**Analysis:**

Images and MSE for different values of K :

From below graph of MSE vs K we can see that mse between actual and clustered image decreases as we increases k and from this observation image should become more clear as we increase K value.



K=2: MSE: 0.03206



K=5 MSE: 0.009577

****

K=10MSE: 0.005056

****

K=20 MSE: 0.002755



K=50 MSE: 0.00128



**Conclusion:**

In the given images, we can observe that as the value of K increases, the number of clusters also increases, and the clarity of the image also improves. This is because, with an increase in the number of clusters, the pixels with similar color values are grouped together, making the image more meaningful and colors more vibrant. Therefore, it is evident that by increasing the number of clusters, the algorithm can capture more details.