Assignment 2

Computer Vision

(Ratio Cut and K-Means Clustering)

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Submitted By:

Gyanendra Chaubey

MTech (AR-VR)

M23AIR005

Submitted To:

Dr. Rajendra Nagar

Assistant Professor,

Dept. of EE, IIT, Jodhpur

Objective:

There are two tasks included in this assignment. First, task was to use the Ratio-Cut based clustering technique to segment the images and then compare it with K-Means clustering technique for cluster numbers as 3 and 6. Then we must compare the results.

Task 1: Ratio Cut Clustering

Ratio Cut Clustering has been implemented for the two images. First, preprocess of the images has been done to resize them into 64x64x3 pixels so that dimension of the image size reduces, and less memory is used for computation.

Following steps has been followed for computing clusters using the Ratio-Cut Method:

1. **Construct the similarity graph**
   Compute pairwise similarity between the pixels based on the features. Construct a similarity matrix aka adjacency matrix W, $W_{i,j}$ represents the similarity between the pixel i and pixel j.
   $$W_{i,j} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)$$
   where σ is a parameter controlling the width of Gaussian Kernel.
2. **Compute the degree matrix D**
   After calculating the similarity, calculate the degree matrix D where $D_{i,i}$ represents sum of weights of edges incident to vertex v_i.
   $$D_{i,i} = \sum_j W_{i,j}$$
3. **Compute the Laplacian Matrix**
   Then compute the Laplacian matrix, from which eigen vectors need to be calculated.
   $$L = D - W$$
4. **Compute the eigen values and eigen vectors**
   Compute the eigen values and eigen vectors of the Laplacian matrix L. Select k smallest eigen vectors corresponding to k smallest eigen values.
   $$Lv = \lambda v$$

   Here λ is the eigen value and v is the eigen vector.

5. **Select the k smallest Eigen Vectors**
   Select k smallest Eigen vectors corresponding to k min eigen values.
   $$H = Eigenvectors[:, [0: N\_clusters\ ]$$
6. **Cluster Eigen vectors using KMeans**
   Cluster selected eigen vectors using the k-means algorithm with k-clusters. Cluster labels are received.
7. **Compute Silhouette Score**
   After computing the cluster labels, Silhoutte score has been calculated for the cluster number 3 and 6 and has been attached in the clustered image display title as can be seen in the results in Fig. 2.
   $$Silhoutte\ Score = \frac{b - a}{\max\left(a, b\right)}$$

Here, a is the mean intra-cluster distance (average distance between a data point and other data points in the same cluster) and b is the mean nearest-cluster distance (average distance between a data point and data points in the nearest cluster that the data point is not a part of).

Image plots of results are shown below:

cluster number = 3 (size 64x64)

cluster number = 6 (size 64x64)

Code Link Task 1: [Click Here]

Task 2: K-Means Clustering

In this work, K-Means clustering has been done for the images. For this, preprocessing of the images has been done to resize them into 64x64x3 pixels so that dimension of the image size reduces, and less memory is used for computation.

Following steps has been followed for computing clusters using the K-Means Method:

1. **Randomly Initialize Centroids**
   The centroids are initialized randomly by selecting k data points from the dataset without replacement. This step ensures that the centroids are spread across the feature space.
2. **Assign Data points to nearest centroid**
   For each data point, the distance to each centroid is computed. The data point is then assigned to the nearest centroid based on these distances. The mathematical formula used here is the Euclidean distance:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3. **Update Centroids**
   After assigning data points to centroids, the centroids are updated by computing the mean of the data points in each cluster. This step ensures that the centroids move towards the center of their respective clusters.

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ji}$$

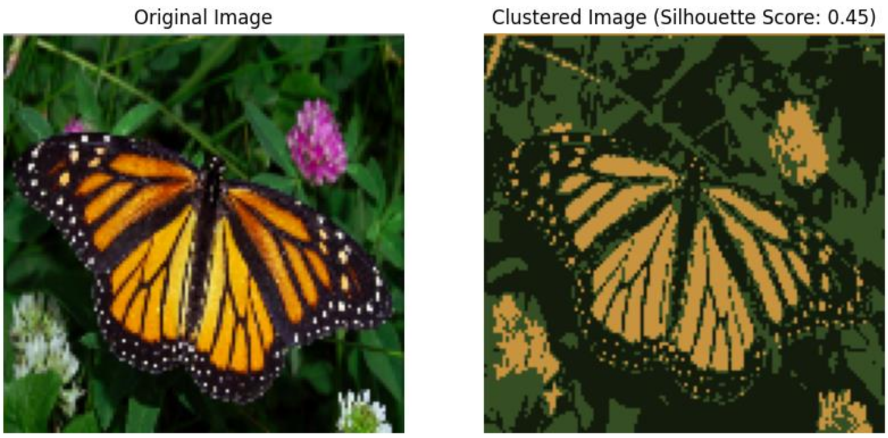
4. **Apply K-Means and Obtain cluster canters and labels**
   K-means clustering is the iterative process of assigning data points to centroids and updating centroids until convergence. The steps involved here are assigning clusters (Step 2) and updating centroids (Step 3). This is repeated until the centroids stop updating it position.
5. **Compute Silhouette Score**
   After computing the cluster labels, Silhoutte score has been calculated for the cluster number 3 and 6 and has been attached in the clustered image display title as can be seen in the results in Fig. 2.

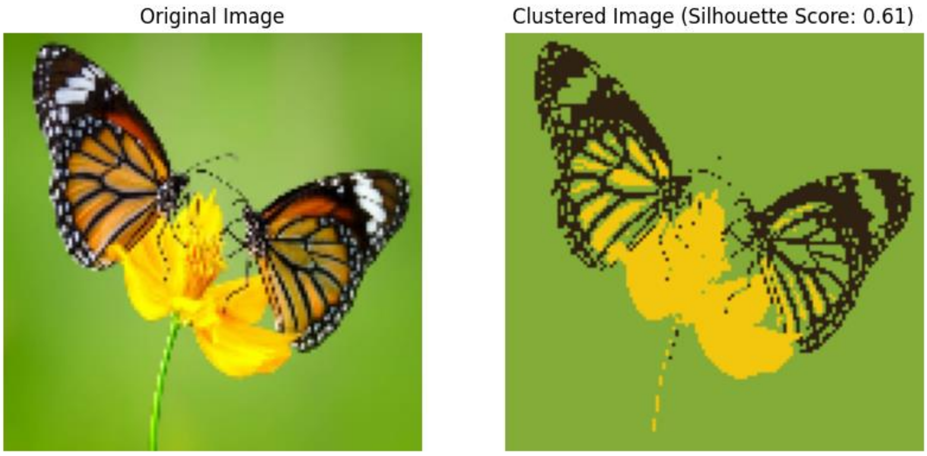
$$Silhoutte\ Score = \frac{b - a}{\max\,(a, b)}$$

   Here, a is the mean intra-cluster distance (average distance between a data point and other data points in the same cluster) and b is the mean nearest-cluster distance (average distance between a data point and data points in the nearest cluster that the data point is not a part of).

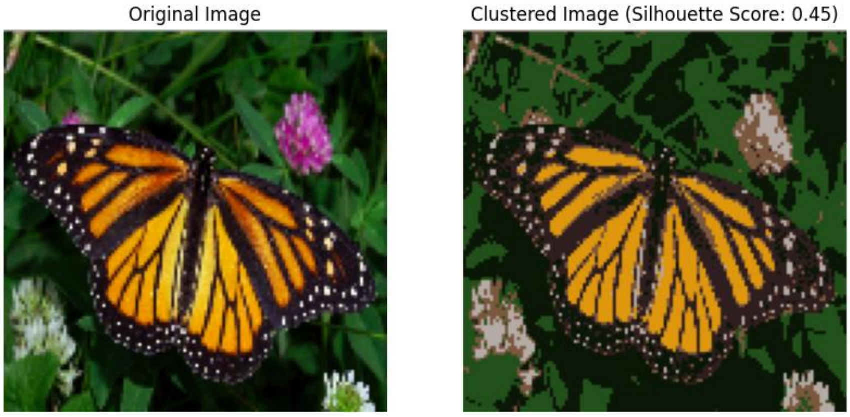
Output of K-Means:

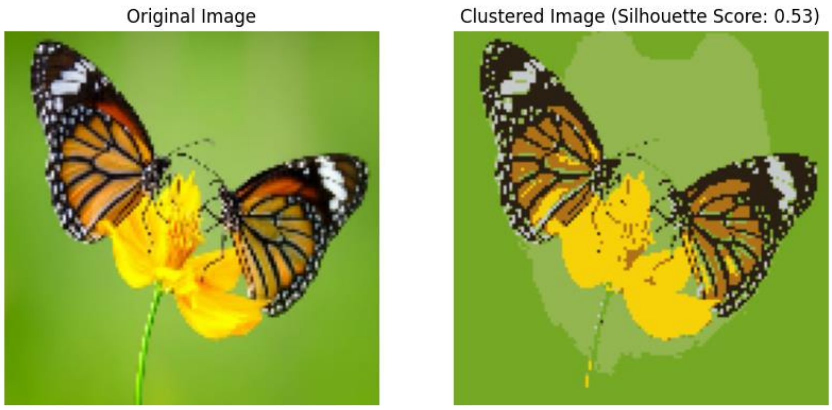
cluster number = 3 (size 128x128)





cluster number = 6 (size 128x128)





cluster number = 3 (size 64x64)

cluster number = 6 (size 64x64)

Comparison of Results (Silhouette Score) for both Ratio-Cut and K-Means

	Cluster = 3	Cluster = 6
Image 1 (Ratio Cut)	0.57	0.45
Image 2 (Ratio Cut)	0.53	0.25
Image 1 (K-Means)	0.45	0.45
Image 2 (K-Means)	0.60	0.53

Code Link Task 2: Click here