Yash Maniya
B20CS033

Computer Vision (CSL7360)
Assignment 2

Q1. Segmentation : Your task is to implement the Ratio-Cut based clustering technique and compare it with the K-means clustering technique. Use the number of clusters as 3 and 6 for both techniques. You have to do the following :
[100 Marks]

- Implement the algorithm from scratch. You should not use any in-built functions to directly get the segmentation. You can use the inbuilt function to perform simple k means, get eigenvalues, eigenvectors, etc.
- Resize the size of the images to 64x64 to reduce the computation complexity.
- Make the code modularise to take images from the folder as an input.
- Compare the performance of the two techniques on Image 1 and Image 2, i.e., compare the 4x2 cases.
- The input images are present in the following location : https://bit.ly/cvasg2img

1. Image Segmentation

Image segmentation is a crucial task in computer vision that involves partitioning an image into multiple segments or regions based on certain characteristics such as color, intensity, texture, or edges. The goal of image segmentation is to simplify the representation of an image and make it easier to analyze and interpret. It finds applications in various fields including medical imaging, object recognition, and autonomous driving.

The provided code implements two popular methods for image segmentation : K-Means clustering and Ratio Cut-based segmentation. Both methods aim to partition an image into meaningful regions, but they employ different approaches to achieve this goal.

2. K-Means Clustering Method

K-Means clustering is an unsupervised learning algorithm used for partitioning a dataset into K distinct, non-overlapping clusters. In the context of image segmentation, K-Means clustering is applied to group similar pixels together based on their feature space representation. The algorithm iteratively assigns each pixel to the cluster whose centroid (representative point) is closest to it, and then updates the centroids based on the mean of the pixels assigned to each cluster.

Code Explanation :

a. Initialization:

- Initialize the `KMeansClustering` object with the number of clusters (`num_clusters`).
- Store the number of clusters as an instance variable.

b. Preprocessing:

- Obtain the dimensions of the input image (`image`) using the `shape` attribute.
- Reshape the image into a 2D array of pixel values using `np.reshape`.
- If the `feature_space` parameter is set to "RGB", use the pixel values directly.
- If the `feature_space` parameter is set to "RGBxy", create a mesh grid of x and y positions using `np.meshgrid`. This preserves locality while clustering.
- Concatenate the pixel values with the x and y positions if "RGBxy" feature space is chosen.

c. K-Means Clustering:

- Initialize a KMeans object with the specified number of clusters (`num_clusters`) and fit it to the pixel values with positions.
- Obtain the cluster labels for each pixel using the `labels_` attribute of the fitted KMeans object.
- Get the cluster centers using the `cluster_centers_` attribute of the fitted KMeans object.
- Use the cluster labels to assign each pixel the corresponding cluster center value.
- Reshape the resulting array to the original image dimensions.

d. Post-processing:

- Create an `ImageManipulation` object with the segmented image data converted to unsigned integer 8-bit format.
- Calculate the window size for the max frequency filter based on the height of the image.

e. Return:

- Return the result of applying the max frequency filter to the segmented image using the `max_frequency_filter` method of the `ImageManipulation` object.

3. Ratio Cut Based Segmentation

Ratio Cut-based segmentation is a graph partitioning algorithm that aims to minimize the ratio cut, which measures the dissimilarity between different segments of an image. In this method, an adjacency matrix is constructed based on the image's gradient magnitude, and K-Means clustering is initially applied to obtain an initial segmentation. Subsequently, an optimization process is performed to refine the segmentation by iteratively reassigning pixels to clusters to minimize the ratio cut.

Code Explanation :

a. Initialization:

- The class `RatioCutClustering` is defined with an `__init__` method that initializes the number of clusters (`num_clusters`) based on the input parameter.

b. Calculation of Ratio Cut:

- The method `_calculate_ratio_cut` calculates the ratio cut for a given set of labels and an adjacency matrix.
- It iterates over each cluster and computes the ratio cut by summing the edge weights between the current cluster and the rest of the clusters.
- The ratio cut is normalized by dividing it by the total number of clusters.

c. Creation of Adjacency Matrix:

- The method `_create_adjacency_matrix` creates an adjacency matrix for the input image.
- It converts the image to grayscale and computes gradients in the x and y directions.
- Edge weights are calculated as the magnitude of gradients, and these values are converted to similarity values using the exponential function.
- The resulting matrix represents the similarity between pixels in the image.

d. Fitting the Model:

- The `fit` method is used to perform the clustering algorithm.
- If the input is not already a NumPy array, it is converted to one.
- K-Means clustering is applied to the pixel values to obtain initial cluster labels.
- An adjacency matrix is created based on the grayscale image.
- The algorithm iteratively optimizes the clustering by randomly reassigning labels and updating them if a lower ratio cut is achieved.
- After a specified number of iterations, the best labels are selected.
- If no improvement is achieved, the labels from the initial K-Means clustering are used.
- Finally, the segmented image is generated by assigning original colors based on the cluster labels.

e. Return:

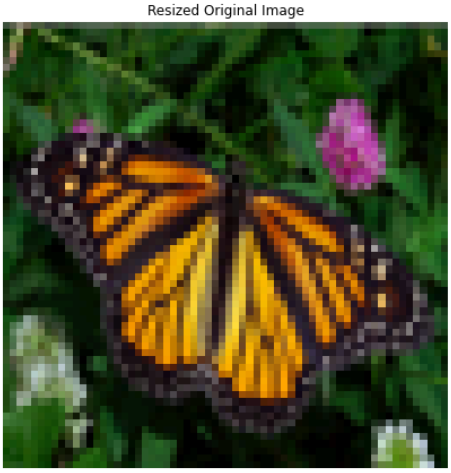
- The method returns the segmented image after applying a maximum frequency filter to enhance the visual quality.
- The window size for the filter is determined based on the height of the image.

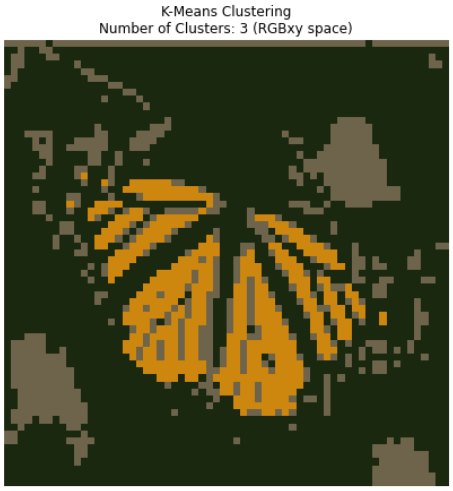
4. Segmentation Results

We have segmented the resized image using both the techniques and used following parameters :

1. `num_clusters` : The number of clusters used for clustering.
2. `feature_space` : The feature space used for clustering, which can be either "RGB" or "RGBxy".

Butterfly Image segmentation results :





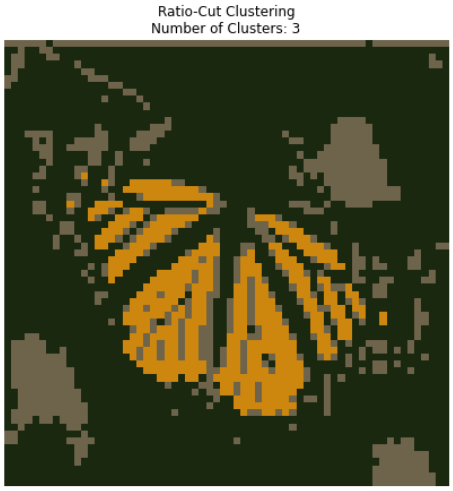
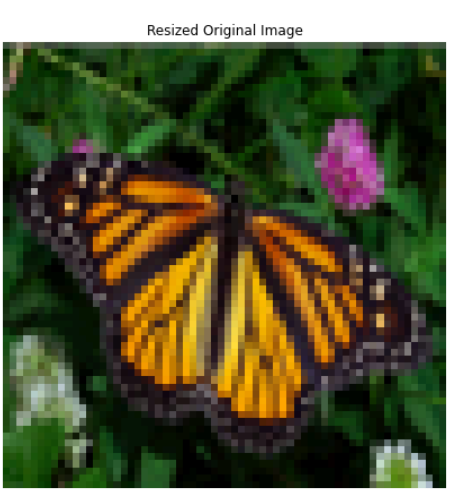
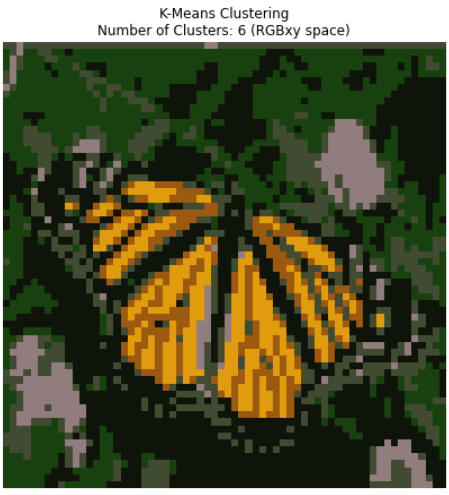


Image Size	Number of Clusters
64 x 64	3





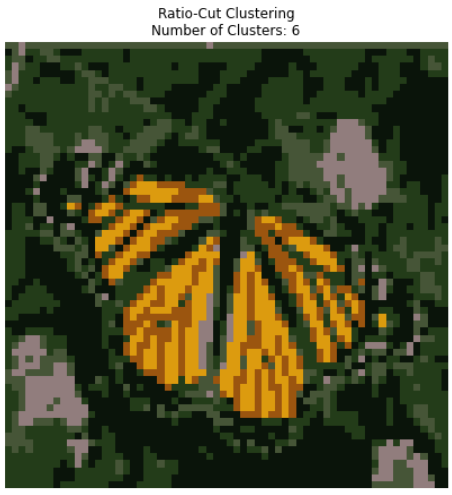
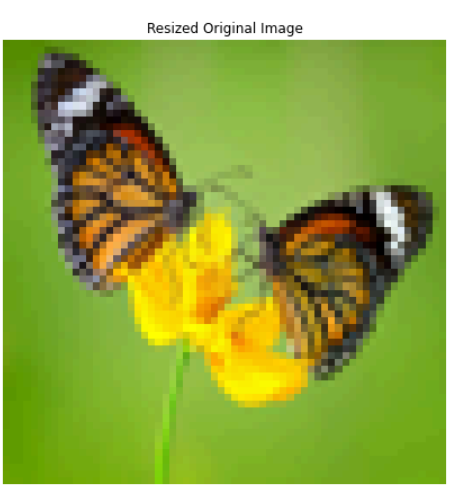
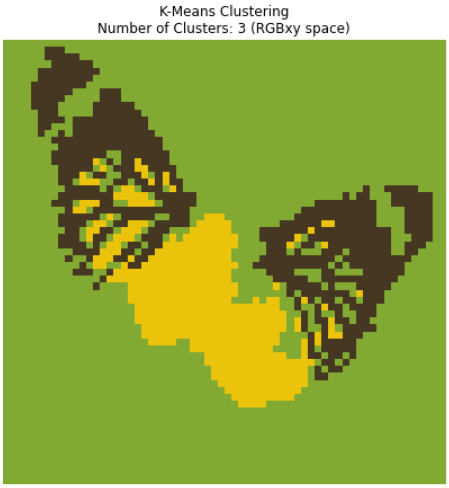


Image Size	Number of Clusters
64 x 64	6





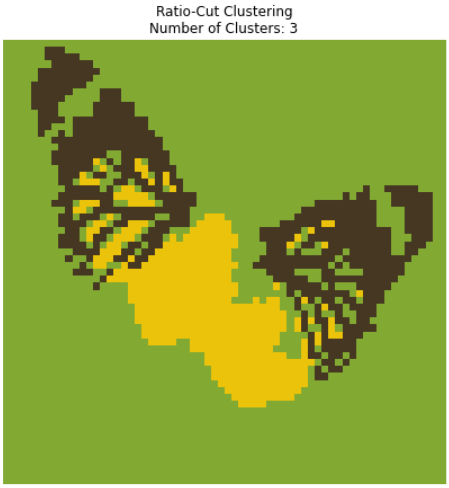
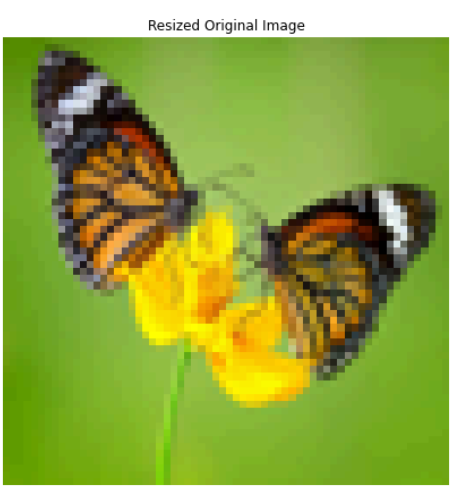
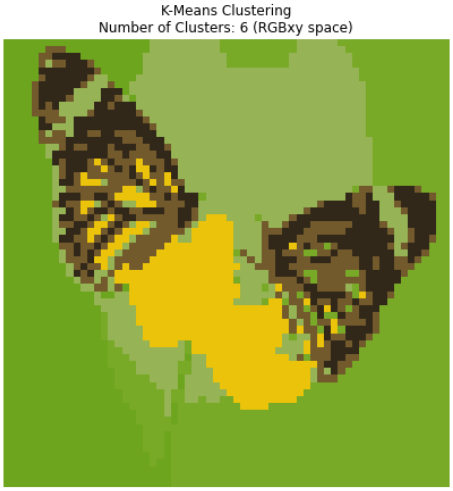


Image Size	Number of Clusters
64 x 64	3





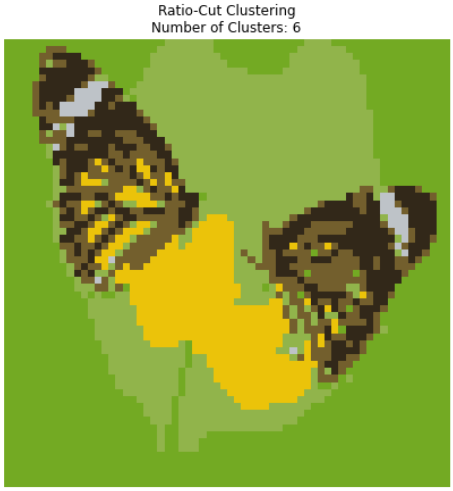


Image Size	Number of Clusters
64 x 64	6

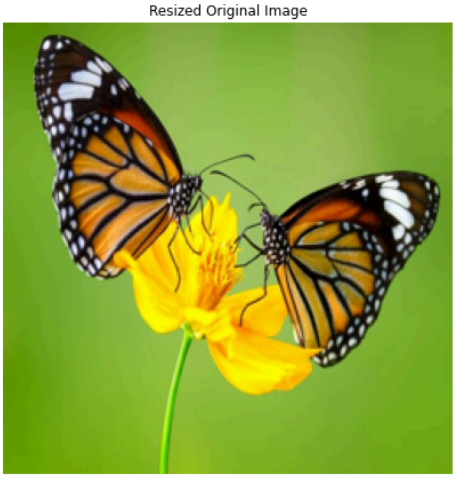
Resized Original Image

K-Means Clustering
Number of Clusters: 6 (RGBxy space)

Ratio-Cut Clustering
Number of Clusters: 6

Image Size	Number of Clusters
256 x 256	6

Resized Original Image



K-Means Clustering
Number of Clusters: 6 (RGBxy space)

Ratio-Cut Clustering
Number of Clusters: 6

Image Size	Number of Clusters
256 x 256	6

Vegetable (Extra Test) Image segmentation results :

Image Size	Number of Clusters
256 x 256	30

Flowers (Extra Test) Image segmentation results :

Image Size	Number of Clusters
256 x 256	10

5. Observations and Conclusion

- Both K-Means clustering and Ratio Cut-based segmentation produce visually appealing segmentation results, but they have different strengths and weaknesses.
- K-Means clustering is simple and computationally efficient but may struggle with complex image structures and textures.
- However K-Means trained on 5 dimensional feature space (`RGBxy`) gives satisfactory results.
- Ratio Cut-based segmentation considers spatial coherence and can handle complex images better but requires more computational resources.

In conclusion, image segmentation is a fundamental task in computer vision with various applications. The choice between K-Means clustering and Ratio Cut-based segmentation depends on the specific requirements of the application, such as computational constraints, the complexity of the images, and the desired level of segmentation quality. By understanding the principles and characteristics of these segmentation methods, one can effectively apply them to different image analysis tasks.