Indian Institute of Technology Jodhpur

Programming Assignment: Image Segmentation

Submitted by

MITESH KUMAR (M23MAC004)

Cloab Link M23MAC004_qu1.ipynb

This assignment compares the result of two images using Kmeans image segmentation and Ratio cut image segmentation.

Methodology

Image Loading and Preprocessing Function

Implemented load_image function to load and preprocess images from the folder. This function performs the following steps:

1. Input: The function takes a folder path as input, where the images are located.

2. Loading Images: It reads the filenames of all images in the specified folder using

3. Resizing: Each image is resized to a fixed size of 64x64 pixels using OpenCV's

4. Colour Space Conversion: As OpenCV reads images in the BGR (Blue-Green-Red) colour format by default, the function converts each image to the RGB (Red-Green-Blue) colour format using

5. Normalisation: After resizing and colour conversion, the pixel values of each image are normalised to the range [0, 1] by dividing by 255.0.

6. Output: The function returns a list of numpy arrays, where each array represents an image with dimensions 64x64 pixels and pixel values normalized to the range [0, 1].

K-Means Clustering Function

Implemented k-means clustering function for performing K-means clustering on an input image. This function performs the following steps:

1. Input: The function takes two arguments, image and number of clusters
2. Flattening Image: The function first flattens the input image into a 2D array where each row represents a pixel and its colour channels.
3. Clustering: It then applies the K-Means algorithm using the KMeans class from sci-kit-learn
4. Fitting and Prediction: The K-Means algorithm is fitted to the flattened image data, and cluster assignments are predicted for each pixel.
5. Segmentation: The cluster centres obtained from K-Means clustering represent the dominant colours in the image. The function assigns each pixel to its nearest cluster centre and reconstructs the segmented image using these cluster centres.
6. Output: The function returns the segmented image, where each pixel is assigned the colour of its nearest cluster centre.

Ratio cut Clustering Function

This function is responsible for performing ratio cut segmentation. This function performs the following steps:

1. Input Parameters:
    - image: The input image to be segmented.
    - num_clusters: The number of clusters to be formed.
    - sigma: The parameter for the Gaussian kernel in computing pairwise distances.
    - lambda: The weight parameter for spatial distance in the pairwise distance computation.

2. Image Preprocessing: The input image is reshaped into a flattened array, where each row represents a pixel in the image.

3. Distance Computations:
    - Two types of distances are computed: intensity distance and spatial distance.
    - Intensity distance: Pairwise Euclidean distances between pixel intensities.
    - Spatial distance: Pairwise distances between pixel coordinates, considering the image grid structure.
    - The total pairwise distance matrix is calculated by combining intensity and spatial distances, with spatial distance weighted by lambda.

4. Adjacency Matrix: The affinity matrix is computed using the exponential of the negative total pairwise distance divided by a constant (sigma) squared. And then normalize it.

5. Laplacian Matrix: The degree matrix and Laplacian matrix are computed from the normalized affinity matrix.

6. Eigenvector Decomposition: Eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix are computed.

7. Clustering: K-means clustering is performed on the selected eigenvectors, with the number of clusters specified by num_clusters.

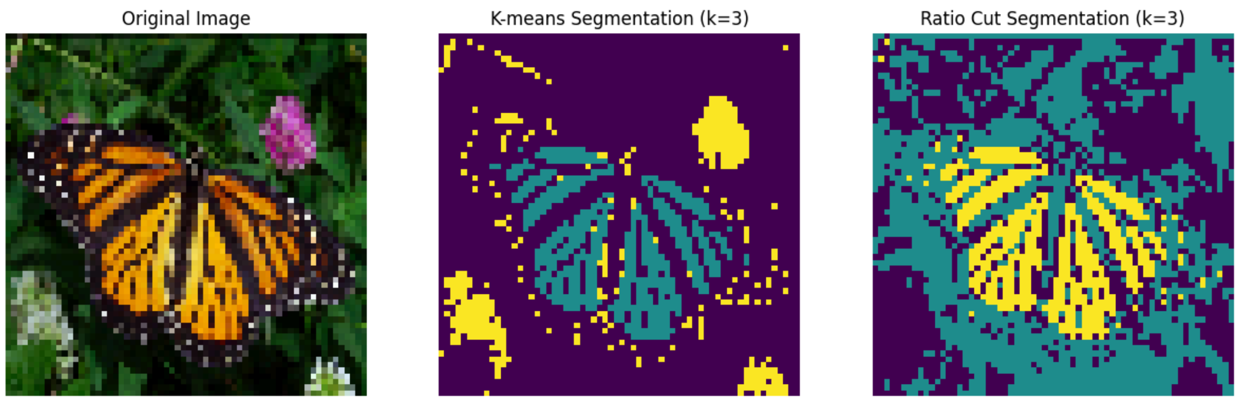
8. Segmentation: The cluster labels obtained from K-means clustering are reshaped to match the original image dimensions, resulting in a segmented image.

9. Output: The function returns the segmented image, where each pixel is assigned a label corresponding to its cluster.

Result comparison between K-means and Ratio cut segmentation

Image1

K = 3, lamda = 1 , sigma = 0.0001



K = 6, lamda = 1 , sigma = 0

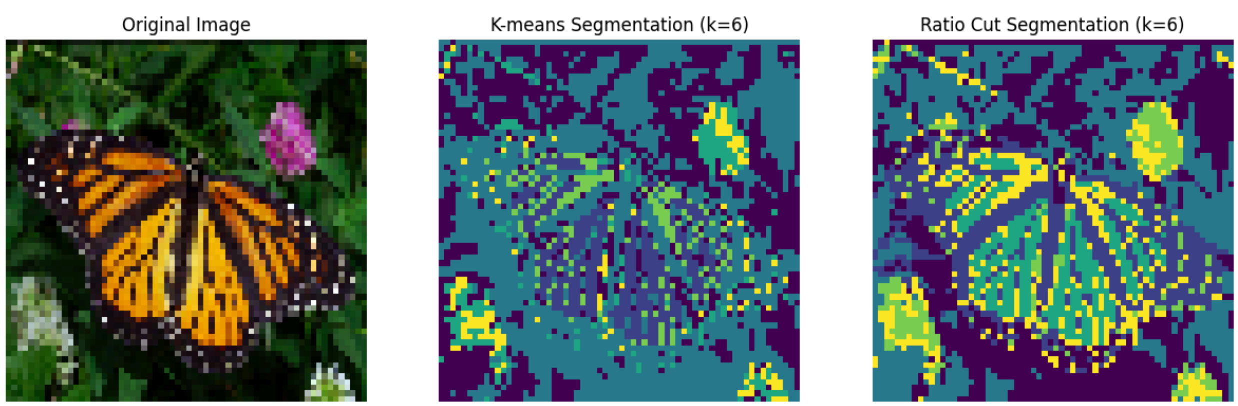
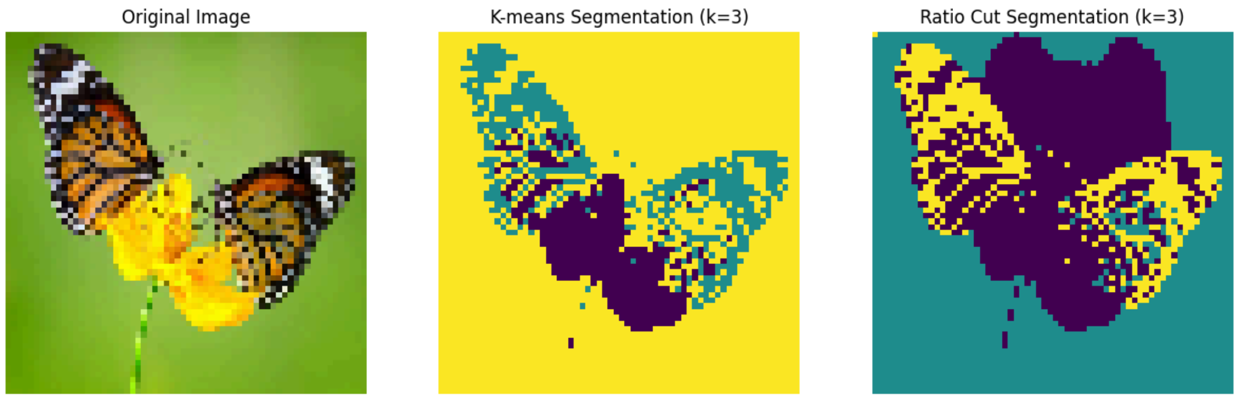
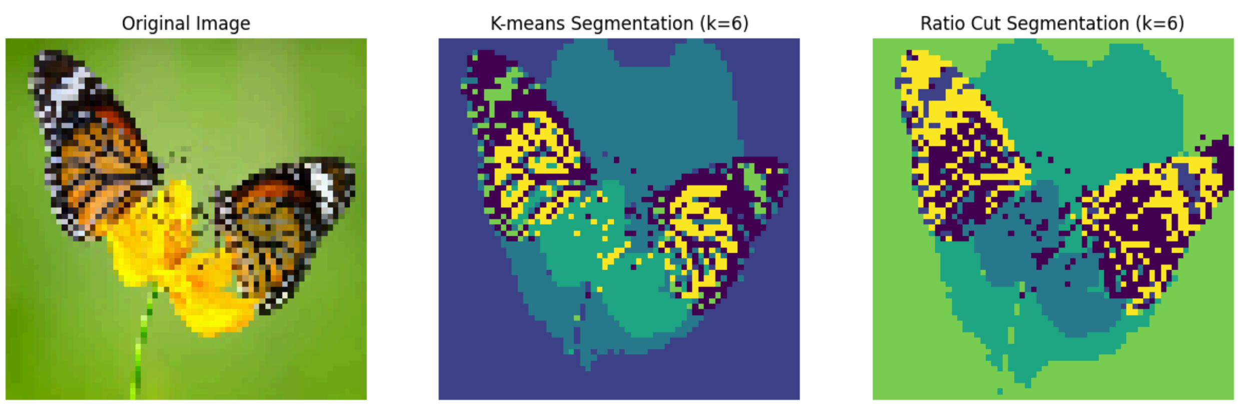


Image2

K $=3$, lamda $=1$, sigma $=0.001$



K $=6$, lamda $=1$, sigma $=0$



Observations:

On visual inspection of the obtained result. It can be concluded that

- Image1 , $k = 3$ →KMeans result is better
- Image1, $k = 6$→Ratio cut result is better
- Image2 , $k = 3$ →Ratio cut result is better
- Image2, $k = 6$→Both results are similar

- Obtained segmented images from both algorithms are similar. Cluster patterns match for both images. While there is a difference in boundary smoothness.

- K-means focuses on minimizing the squared distances between data points and their assigned cluster centroids. Consequently, it tends to create compact and uniform clusters with well-defined boundaries.

- On the other hand, Ratio-Cut clustering seeks to minimize the normalized cut criterion, which considers the similarity between neighbouring pixels. This objective can lead to the formation of clusters with more irregular shapes that better capture the underlying structure of the image.