Experiment-1.3

Aim of the Experiment : Image segmentation using Mean-Shift and Graph-Cut Algorithms

1. Image Segmentation using Mean-Shift algorithm
2. Image Segmentation using Graph-Cut algorithm

I. Mean Shift Segmentation Method

Problem Description :

Mean Shift is a non-parametric clustering algorithm commonly used in image processing and computer vision, including image segmentation. Mean Shift segmentation is a technique that identifies homogeneous regions or segments within an image based on the similarity of pixel values. Here's a brief overview of the Mean Shift segmentation process:

Kernel Density Estimation:

Mean Shift starts by treating each pixel in the image as a data point in a high-dimensional space, where the dimensions represent the pixel values in different color channels. A kernel function is then applied to each data point, creating a probability density function (PDF) that represents the likelihood of finding other data points in the vicinity.

Mean Shift Iteration:

Mean Shift involves an iterative process where each data point (pixel) is shifted towards the mode (peak) of the estimated probability density. The shift is determined by calculating the mean of the data points within a certain neighborhood defined by the kernel. The process is repeated until convergence, where the data points no longer move significantly.

Segmentation:

After convergence, data points that have shifted to the same mode are considered part of the same segment or cluster. The segmentation result is obtained by assigning a label to each pixel based on the cluster to which it converges.

Parameter Tuning:

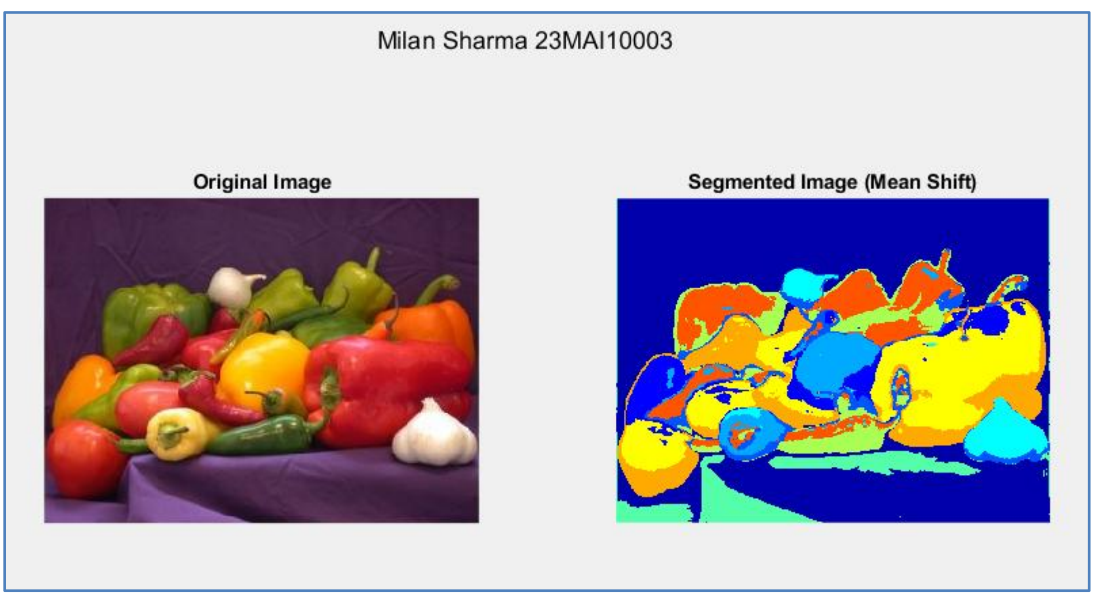
Mean Shift has a bandwidth parameter that influences the size of the neighborhood considered during the mean shift iteration. Adjusting this parameter can impact the segmentation results.

Mean Shift segmentation is effective in capturing complex structures and adapting to different shapes and sizes of regions within an image. It doesn't require specifying the number of clusters in advance, making it suitable for applications like image segmentation, where the number of segments may not be known beforehand.

Code:

```
%Read an image from file
originalImage = imread('C:\Users\milan\Downloads\fruits.jpg');

% Display the original image
subplot(1,2,1), imshow(originalImage);
title('Original Image');


% Convert the image to Lab color space
labImage = rgb2lab(single(originalImage));

% Reshape the image for mean shift
reshapedImage = reshape(labImage, [], 3);

% Perform mean shift clustering
[clusterIndices, ~] = imsegkmeans(labImage, 10, 'NumAttempts', 3);

% Reshape the cluster indices to the original image size
segmentedImageMeanShift = reshape(clusterIndices, size(originalImage, 1),
size(originalImage, 2));

% Display the segmented image using mean shift

subplot(1,2,2), imshow(label2rgb(segmentedImageMeanShift));
title('Segmented Image (Mean Shift)');

sgtitle("Milan Sharma 23MAI10003");
```

Output :



II. Graph Cut Segmentation Method

Problem Description :

Graph cut segmentation is a computer vision and image processing technique used for image segmentation. It's based on the graph theory concept of graph cuts and is particularly useful for dividing an image into meaningful regions or objects. One common algorithm for graph cut segmentation is the min-cut/max-flow algorithm. The idea is to model the image as a graph, where pixels are nodes, and edges between pixels represent relationships based on similarity or dissimilarity. Here's a simplified overview of the graph cut segmentation process:

Graph Construction:

Each pixel in the image is represented as a node in the graph. Edges between nodes are assigned weights based on the dissimilarity or similarity between pixel values. This could involve color similarity, intensity differences, or other features depending on the application. Two special nodes, a source and a sink, are added to the graph. These represent the background and foreground or the two segments being sought.

Capacity Assignments:

The capacities of the edges between nodes are determined based on the dissimilarity/similarity between pixel values. Higher weights indicate stronger connections.

Min-Cut/Max-Flow Algorithm:

The min-cut/max-flow algorithm is applied to find the cut in the graph that minimizes the total capacity of the cut. This cut effectively separates the graph into two disjoint sets of nodes: the source side (representing one segment) and the sink side (representing the other segment).

The min-cut algorithm works by finding the path with the minimum total capacity from the source to the sink in the graph.

Segmentation Result:

The cut obtained by the algorithm defines the segmentation result. Nodes on one side of the cut belong to one segment, while nodes on the other side belong to the second segment.

Graph cut segmentation is powerful and can handle complex image structures and irregular shapes. It allows for incorporating various cues, such as color, texture, or gradient information, into the segmentation process. However, it may be sensitive to the choice of parameters and may not be as efficient for very large datasets. Popular variations of graph cut segmentation include the normalized cut algorithm and the random walker algorithm, each with its own strengths and applications.