PROJECT REPORT

Phase 01

Domain: MATLAB Programming.

Objectives: Image Segmentation.

What is Image Segmentation?

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in colour or shape.

Image segmentation fundamentals

Segmentation subdivides an image into its constituent regions or objects, until the objects of interest in an application have been isolated.

• Segmentation partitions image R into subregions R1, R2, …, Rn such that:

- $R1 \cup R2 \cup \ldots \cup Rn = R$
- Each Ri is a connected set, i = 1, 2, ….., n
- $Ri \cap Rj = \emptyset$ for all i, j where i # j
- Q(Ri)=TRUE for every i
- $Q(Ri \cup Rj)$= FALSE for any two adjacent regions

• Segmentation problem: to partition the image into regions satisfying above conditions.

Two principal approaches:

- Edge-based Segmentation

- Partition an image based on abrupt changes in intensity(edges).

- Region-based Segmentation

- Partition an image into regions that are similar according to a set of pre-defined criteria.

Why Image segmentation is Important?

Several algorithms and techniques for image segmentation have been developed over the years using domain-specific knowledge to effectively solve segmentation problems in that specific application area. These applications include medical imaging, automated driving, video surveillance, and machine vision.

How Image segmentation works?

Image segmentation involves converting an image into a collection of regions of pixels that are represented by a mask or a labelled image. By dividing an image into segments, you can process only the important segments of the image instead of processing the entire image.

A common technique is to look for abrupt discontinuities in pixel values, which typically indicate edges that define a region.

Another common approach is to detect similarities in the regions of an image. Some techniques that follow this approach are region growing, clustering, and thresholding. A variety of other approaches to perform image segmentation have been developed over the years using domain-specific knowledge to effectively solve segmentation problems in specific application areas.

Image segmentation in MATLAB

With MATLAB®, you can:

- Use apps to interactively explore different segmentation techniques
- Simplify image analysis workflows using built-in image segmentation algorithms
- Perform deep learning for image segmentation

Using Apps to Interactively Threshold Images

Image Segmenter App:

Using the interactive Image Segmenter app, you can iteratively try several methods to segment an image before achieving the desired result. For example, you can use the app to segment and further refine the results of an MRI image of a knee with different methods.

Color Thresholder App:

This Color Thresholder app lets you apply thresholding to color images by manipulating the color of the images interactively, based on different color spaces. For example, you can use the Color Thresholder app to create a binary mask using point cloud controls for a color image.

Using a Variety of Image Segmentation Techniques

With functions in MATLAB and Image Processing Toolbox™, you can experiment and build expertise on the different image segmentation techniques, including thresholding, clustering, graph-based segmentation, and region growing.

Thresholding:

Using Otsu's method, imbinarize performs thresholding on a 2D or 3D grayscale image to create a binary image. To produce a binary image from an RGB color image, use rgb2gray to first convert it to a grayscale image.

Clustering:

This technique lets you create a segmented labeled image using a specific clustering algorithm. Using K-means clustering–based segmentation, imsegkmeans segments an image into K number of clusters.

Graph-Based Segmentation:

Graph-based segmentation techniques like lazy-snapping enable you to segment an image into foreground and background regions. MATLAB lets you perform this segmentation on your image either programmatically (lazysnapping) or interactively using the Image Segmenter app.

Region Growing:

Region growing is a simple region-based (also classified as a pixel-based) image segmentation method. A popularly used algorithm is active contour, which examines neighbouring pixels of initial seed points and determines iteratively whether the pixel neighbours should be added to the region. You can also perform this segmentation on images using the Image Segmenter app.

- Edges and thresholds sometimes do not give good results for segmentation. Thresholding still produces isolated image.
- Region growing algorithms works on principle of similarity.
- It states that a region is coherent if all the pixels of that region are homogeneous with respect to some characteristics such as colour, intensity, texture, or other statistical properties.
- Thus idea is to pick a pixel inside a region of interest as a starting point (also known as a seed point) and allowing it to grow.
- Seed point is compared with its neighbours, and if the properties match, they are merged together.

- This process is repeated till the regions converge to an extent that no further merging is possible

Region-Growing Algorithm:

- It is a process of grouping the pixels or subregions to get a bigger region present in an image.
- Selection of the initial seed: Initial seed that represent the ROI should be given typically by the user. Can be chosen automatically. The seeds can be either single or multiple.
- Seed growing criteria: Similarity criterion denotes the minimum difference in the grey levels or the average of the set of pixels. Thus, the initial seed 'grows' by adding the neighbours if they share the same properties as the initial seed.
- Terminate process: If further growing is not possible then terminate region growing process.

Region splitting:

- Entire image is assumed as a single region. Then the homogeneity(similarity) test is applied, where pixels that are similar are grouped together. If the conditions are not met, then the regions are split into four quadrants, else leave the region as it is.
- Split and continue the subdivision process until some stopping criteria is fulfilled. The stopping criteria often occur at a stage where no further splitting is possible.
- This process is repeated for each quadrant until all the regions meet the required homogeneity criteria. If the regions are too small, then the division process is stopped

Region merging:

- Region merging is opposite to region splitting.
- Here we start from the pixel level and consider each of them as a homogeneous region.
- At any level of merging, we check if the four adjacent regions satisfy the homogeneity property. If yes, they are merged to form a bigger region, otherwise the regions are left as they are.
- This is repeated until no further region exits that requires merging

Region splitting and merging:

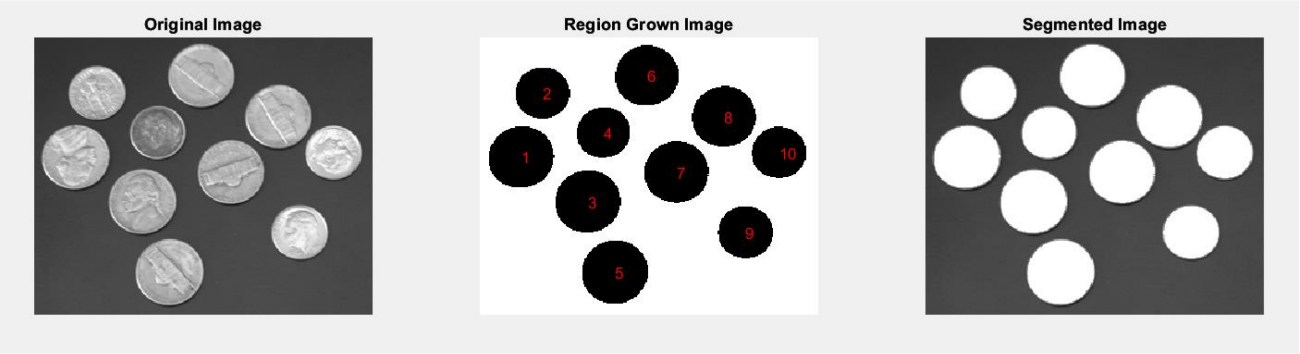
- Splitting or merging might not produce good results when applied separately. Better results can be obtained by interleaving merge and split operations.
- The split and merge procedure is as follows:
- First there is a large region (possible the entire image).

  a) Split into four disjoint quadrants any region Ri for which P(Ri) = FALSE.

  b) Merge any adjacent regions Rj and Rk for which P(Rj U Rk) = TRUE. (the quadtree structure may not be preserved).

c) Stop when no further merging or splitting is possible

Region Splitting and Merging Example:



Deep Learning for Image Segmentation:

Using convolutional neural networks (CNNs), a deep learning technique called semantic segmentation lets you associate every pixel of an image with a class label. Applications for semantic segmentation include autonomous driving, industrial inspection, medical imaging, and satellite image analysis.

Using MATLAB, you can design and train semantic segmentation networks with a collection of images and their corresponding labeled images, and then use the trained network to label new images. To label the training images, you can use the Image Labeler, Video Labeler, or Ground Truth Labeler apps.