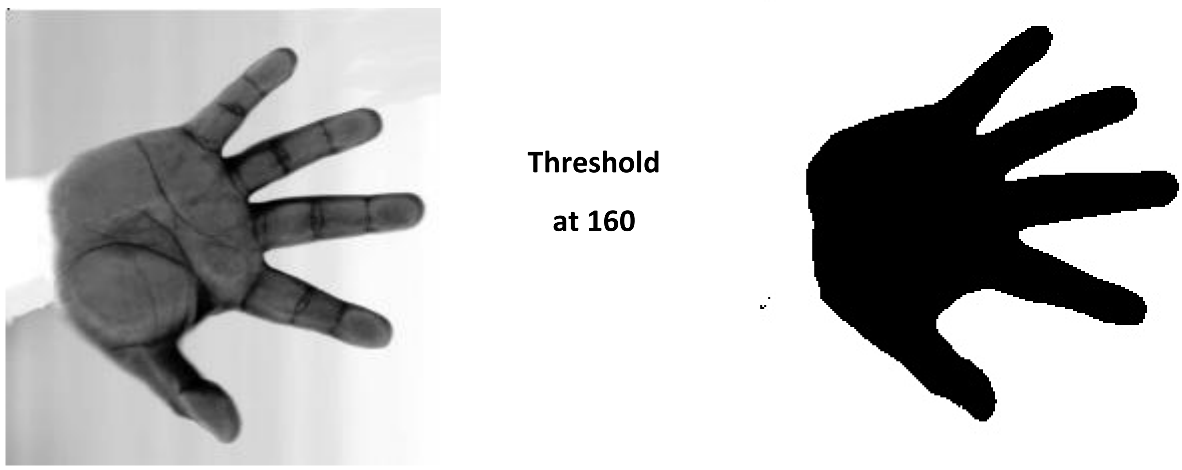
LAB # 06: Image Segmentation

Objective:

The objective of this lab is to apply different segmentation techniques to distinguish different parts of an image.

Theory:

Image Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. For example in the image shown below threshold is applied around the gray value 160 and we have easily extracted the hand out of the image.



Adaptive thresholding:

In some cases a simple threshold does not work. An alternative approach is local threshold which is to statistically examine the intensity values of the local neighborhood of each pixel and threshold on the bases of local mean, median or mode etc.

Local adaptive thresholding:

Local adaptive thresholding is used to convert an image consisting of gray scale pixels to just black and white scale pixels. Unlike the global thresholding technique, local adaptive thresholding chooses different threshold values for every pixel in the image based on an analysis of its neighboring pixels. This is to allow images with varying contrast levels where a global thresholding technique will not work satisfactorily. There are a number of different forms of adaptive thresholding algorithm reported in the image processing literature.

Global adaptive algorithm:

1. Estimate initial mean value of image T1. Make two groups (G1, G2) of pixels based on mean value T1.
2. Compute average gray values m1 and m2 of each group.
3. Compute new threshold value T=(m1+m2)/2
4. Repeat steps 2 to 4 to get abs(T_i-T_{i-1})<epsilon
5. When the above condition satisfies we will have final threshold value depend upon on that we will create binary image.

Lab Tasks:

Lab Task 1:

Write a program that thresholds the provided image (Threshold_Image.png) using global mean and median.

Lab Task 2:

Now threshold the same image by taking threshold value mean and median of 3x3 block locally.

Lab Task 3:

Write a function that takes hematological image as input and segments the image using K-mean clustering with and k=2. Write the function such that the k value can be passed as a parameter. Display the resulting image

Home Task:

Edge Detection Using Canny Edge Detector

In order to determine the edges of an image using the method proposed by J. Canny, the following steps need to be performed on the input image:

1. The first step is to apply a Gaussian smoothing filter on the input image. A Gaussian filter of 5x5 will suffice for the task at hand.
2. The next step is to detect the horizontal and vertical edges in the image using Horizontal Sobel and Vertical Sobel.
3. Once the edges have been computed, the magnitude and the phase need to be computed using the following formulas:

$$Magnitude = \sqrt{(Sobel_x)^2 + (Sobel_y)^2}$$

$$Phase = \tan^{-1}\left(\frac{Sobel_y}{Sobel_x}\right)$$

4. Once the magnitude and the phase have been computed, the next step is to phase quantization. To achieve that, the phase against every pixel should be either 0, 45, 90 or -45 (or 135 instead of -45). If the phase values aren’t in this way then they should be quantized into the following levels: 0, 45, 90, 135. This can be done by computing the closest quantization level of a phase and then giving it that value. For example, if the phase value is 37, then it can be changed to 45. (This step may be unnecessary if the phase values computed already have 4 levels.)
5. Once phase quantization is complete, the next step is to traverse the magnitude array. For each coordinate in the magnitude array, check the value of phase at the same coordinates. Depending on the phase, the (i-1,j-1) and (i+1,j+1) pixels need to be checked in the magnitude array. If the value at (i,j) is less than either one of the (i-1,j-1) and (i+1,j+1) then it is set to 0. Otherwise, its value is preserved. The decision of (i-1,j-1) and (i+1,j+1) depends on the phase. It can be made generic in the following form (let’s assume (i, j) to be (5,5)) :
    a. If phase is 0, then check (5,4) and (5,6).
    b. If phase is 90, then check (4,5) and (6,5).
    c. If phase is 45, then check (4,6) and (6,4).
    d. If phase is -45 (or 135) then check (4,4) and (6,6).
6. The next step is to perform thresholding. Find out the maximum value from the magnitude array after Step 5 has been completed. Using this value, set the pixel values less than or equal to 0.25*Max_value to 0 and set the pixel values greater than or equal to 0.75*Max_value to 1.
7. The next step involves taking a decision for the values that lie in the range (0.25*Max_Value > x < 0.75*Max_Value). Only those pixels values in this range are to be kept that are connected with strong values i.e. connected with the values that have been set to 1 after thresholding. To achieve this, an intermediate array can be obtained in which all the values that lie in the range above have been set to 1 and all else is set to 0. Using this intermediate array and the magnitude array on which two previous thresholds have been applied, it can be checked whether to keep a value or not. For that purpose, for each value in the intermediate array, its eight neighbors can be checked in the thresholded array. If a single one of them is 1, then this value will be kept, otherwise it will be discarded.
8. Once the above step is complete, combining the now processes intermediate array and the thresholded array should give the edge map of the image.