ED6001: Medical Image Analysis

Programming Assignment 3

Name: Mohammad Al Fahim K **Roll No.:**EE21S050

1. **Brief Description:**

    The Histogram based Clustering Estimation (HBCE) algorithm was used to come up with the required number of intensity levels for optimum segmentation. After HBCE, K-Means clustering was performed on the resulting intensity levels of HBCE algorithm to assign the binary class labels.

    Pseudo Code:

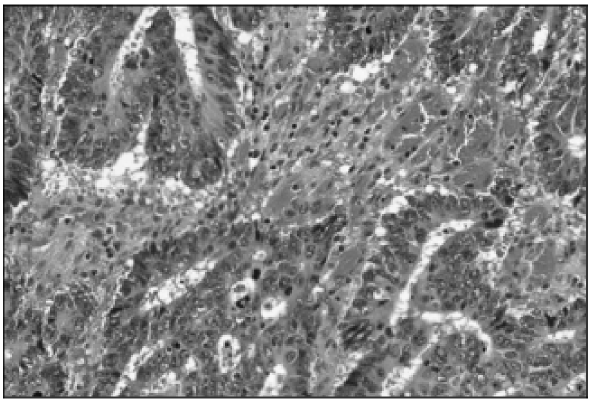
    - img = Input(image)
    - hist = histogram(img)
    - beta_intens, beta_freq = 1, 1 # changes with each image
    - ppv = PeakPitVector(hist) #get the intensities of the peaks and pits in the histogram
    - ld = LD(ppv) #calculates distance between peak and its corresponding pit
    - mean_ld = mean(ld)
    - intensity_threshold = mean(ld)*beta_intens
    - sld = [] #new vector to store result of intensity search
    - for loop over ld:
        - if ld[i]>intensity_threshold: sld.append(peak and pit) #peak and pit corresponding to ld[i]
    - hd = HD(sld) #calculates frequency distance between peak and corresponding pit
    - mean_hd = mean(hd)
    - frequency_threshold = mean(hd)*beta_freq
    - shd=[] #new vector to store result of frequency search
    - for loop over hd:
        - if hd[i]>frequency_threshold: shd.append(peak and pit) #peak and pit corresponding to hd[i]
    - class_labels = Kmeans_classify(shd, n_clusters=2)

    Flowchart:

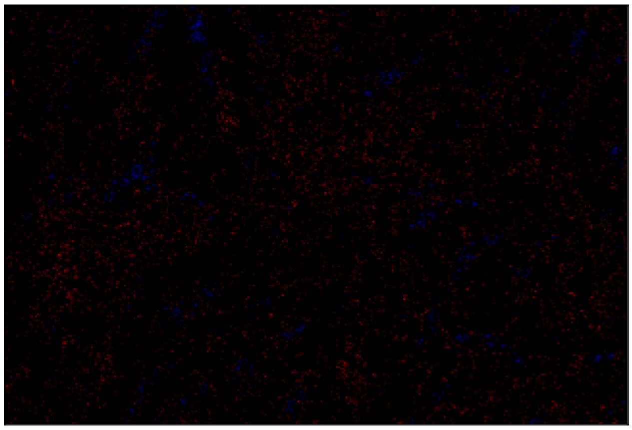
    1. The image is taken as input and its histogram is calculated. The $\beta_{intensity}$ and $\beta_{frequency}$ values are set.
    2. A search algorithm is run which searches for peaks and their corresponding pits in the histogram. The intensities of the pairs of peaks and pits are stored in the ppv vector.
    3. The ld vectors stores the distances between the peak intensities and their corresponding pit intensity. Its mean is then calculated to calculate the intensity_threshold.
    4. The pairs of peak and pit intensities which have a distance greater than intensity_threshold are stored in the sld vector.
    5. The hd vector stores the distance between the frequencies of peak and pit intensities.
    6. Its mean is calculated which is used to calculate the frequency_threshold.
    7. The pairs of peak and pit intensities which have a frequency distance greater than frequency_threshold are stored in the shd vector.
    8. Finally, K-Means classifier initialized with 2 clusters is used to classify the intensities in the shd vector as belonging to the cell structure (foreground) or background.

Results:

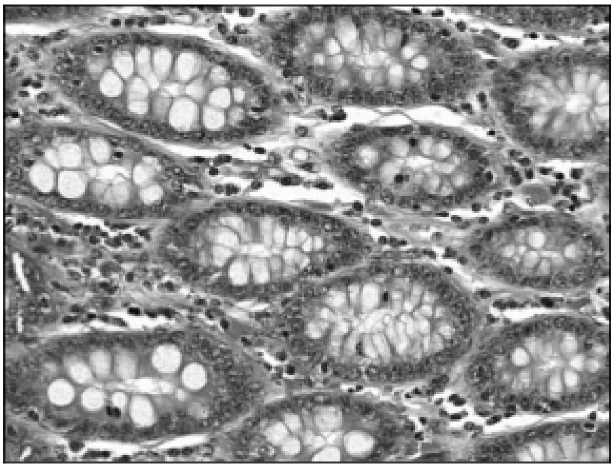
Original image 1



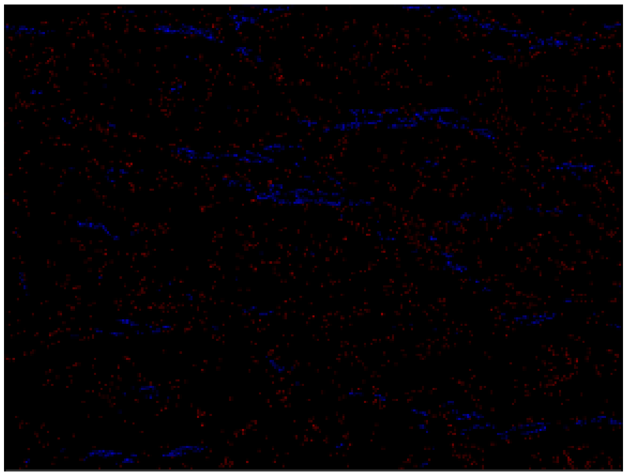
Assigned class labels



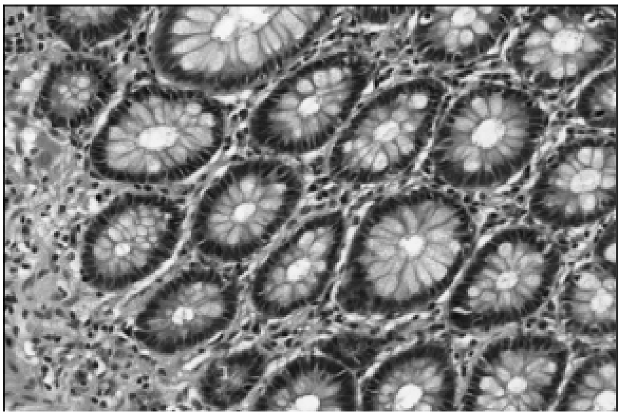
Original image 2



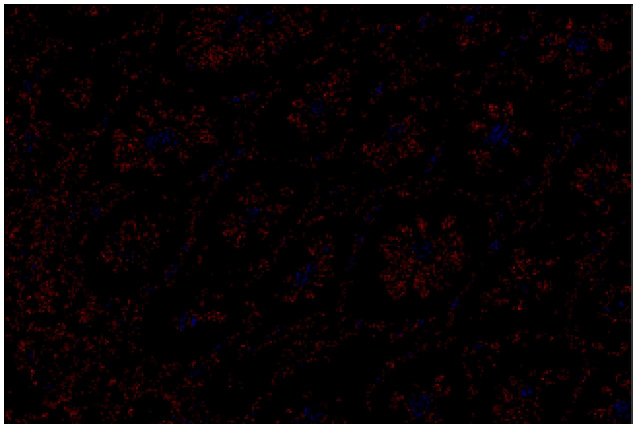
Assigned class labels



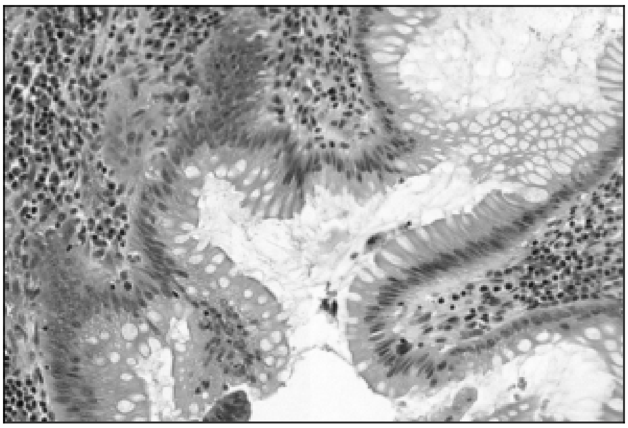
Original image 3



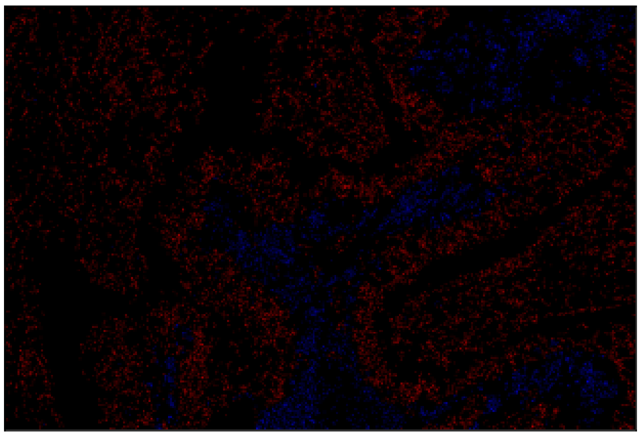
Assigned class labels

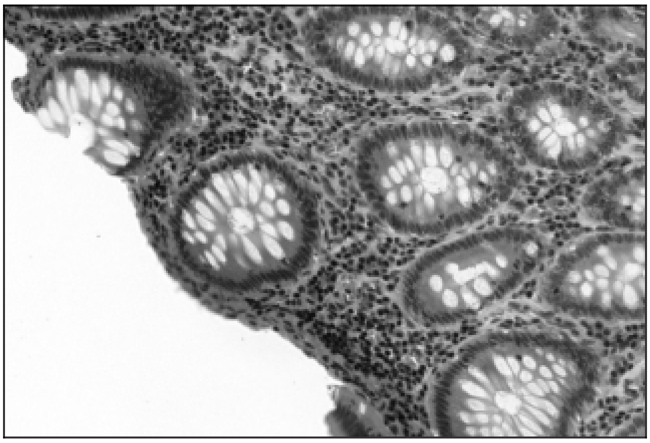


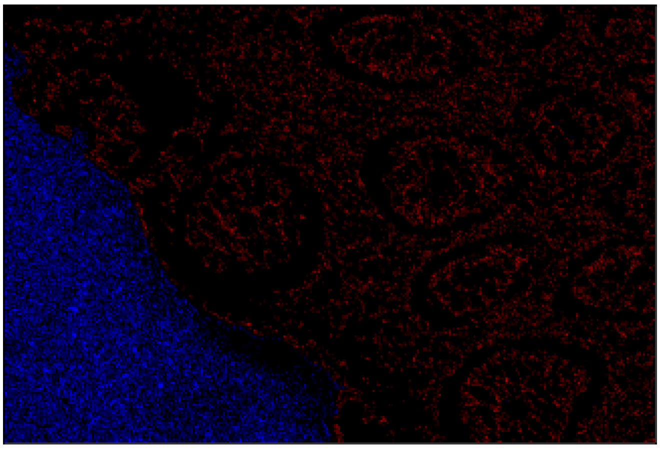
Original image 4



Assigned class labels







Observations:

Although the assigned class labels images seem almost black, deeply looking into it will reveal red and blue regions amidst the black pixels. The red region corresponds to the foreground and the blue region corresponds to the background. The black pixels neither belongs to the foreground nor to the background. For the 1st and 2nd images, the $\beta_{intensity}$ and $\beta_{frequency}$ were both set to 1.3. For the 3rd image, the $\beta_{intensity}$ and $\beta_{frequency}$ were both set to 1.1. For the 4th image, the $\beta_{intensity}$ and $\beta_{frequency}$ were both set to 1. For the 5th image, the $\beta_{intensity}$ and $\beta_{frequency}$ were both set to 0.5. These values were set based on how well the alpha beta swap and alpha expansion methods worked.

2. Brief Description

To apply graph cut method to optimize the delineation of the gland from the background using minimization algorithms: alpha-expansion and alpha-beta swap. The shd vectors from the 1st question are used for the alpha-expansion and alpha-beta swap methods.

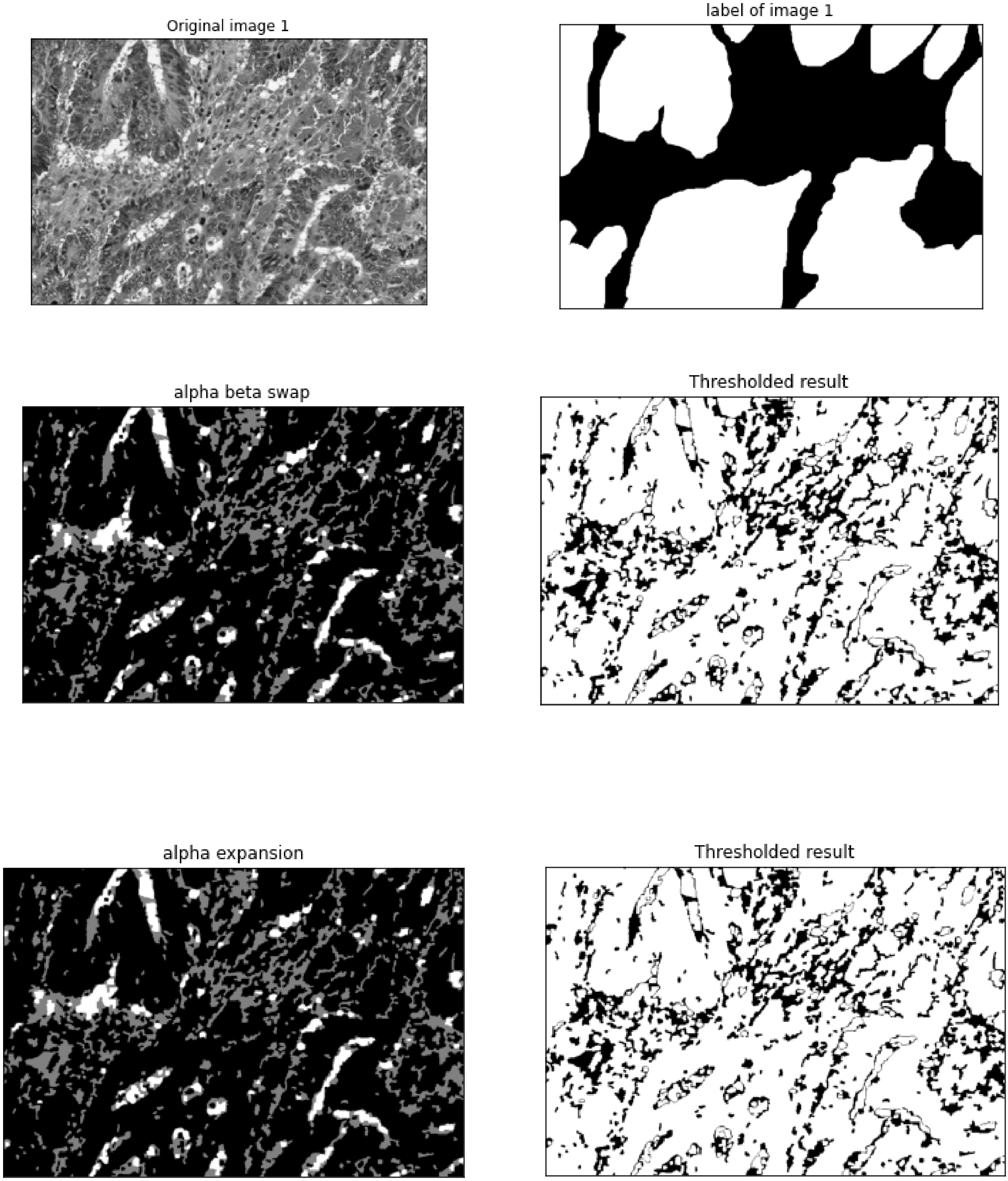
Pseudo Code: (continued from question 1)

- levs = shd/255
- I = img/255
- D = abs(I.reshape(I.shape+(1,)) - levs.reshape((1,1,-1)))
- V = abs(levs.reshape((-1,1)) – levs.reshape((1,-1)))
- label = abswap_grid(D,V) (or) aexpansion_grid(D,V)

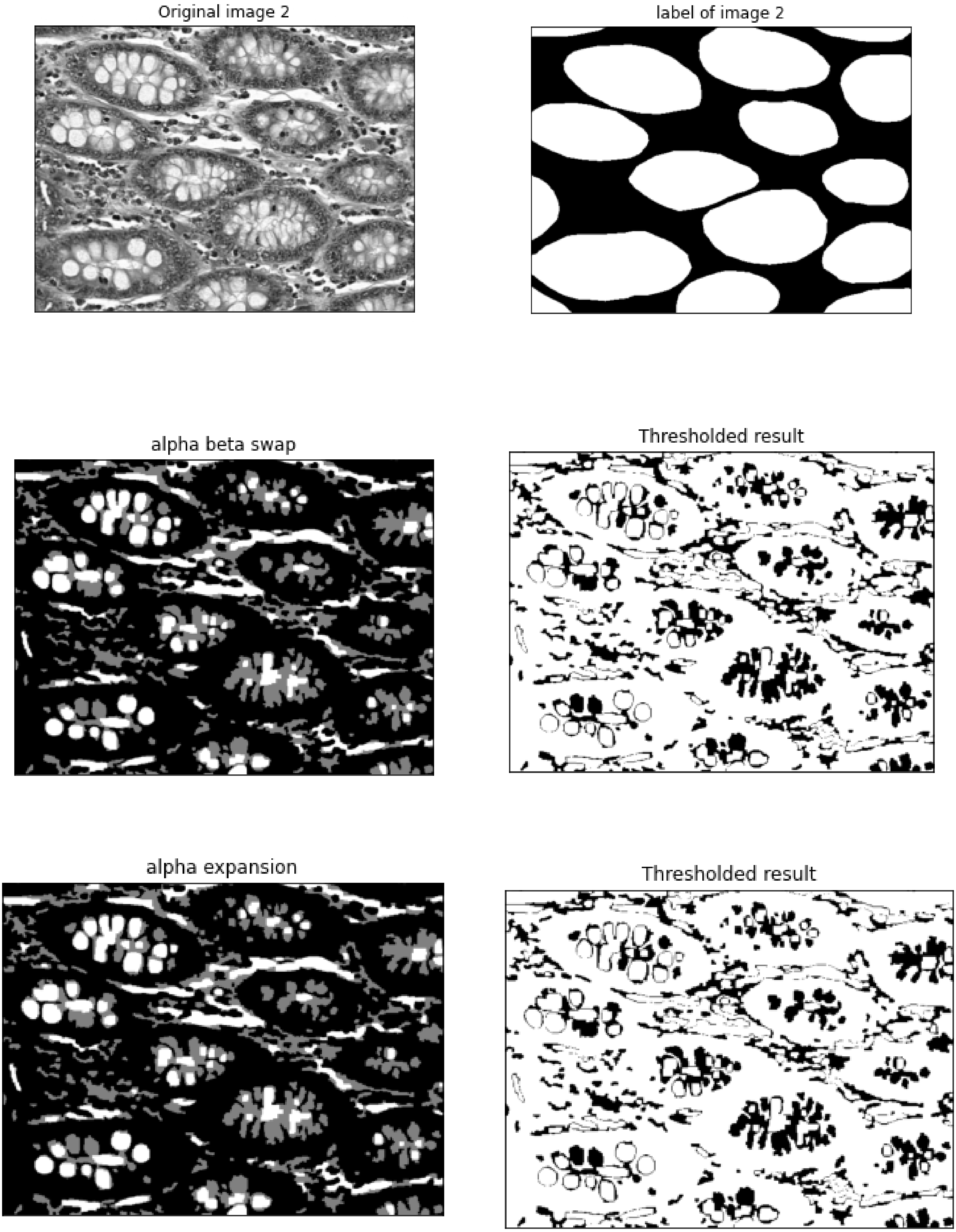
Flowchart:

1. The levs vector and I array are generated by dividing the shd and img arrays by 255
2. The D matrix is calculated as the absolute difference between all image pixels and each of the intensity in the levs vector. Its final shape is (img.shape, len(levs))
3. The V matrix is calculated as the absolute difference between each of the intensities in the levs vector. Its final shape is (len(levs),len(levs)).
4. The D and V matrices are finally used for the abswap_grid() and aexpansion_grid() function for alpha-beta swap and alpha expansion methods.
5. The label is the segmented image.

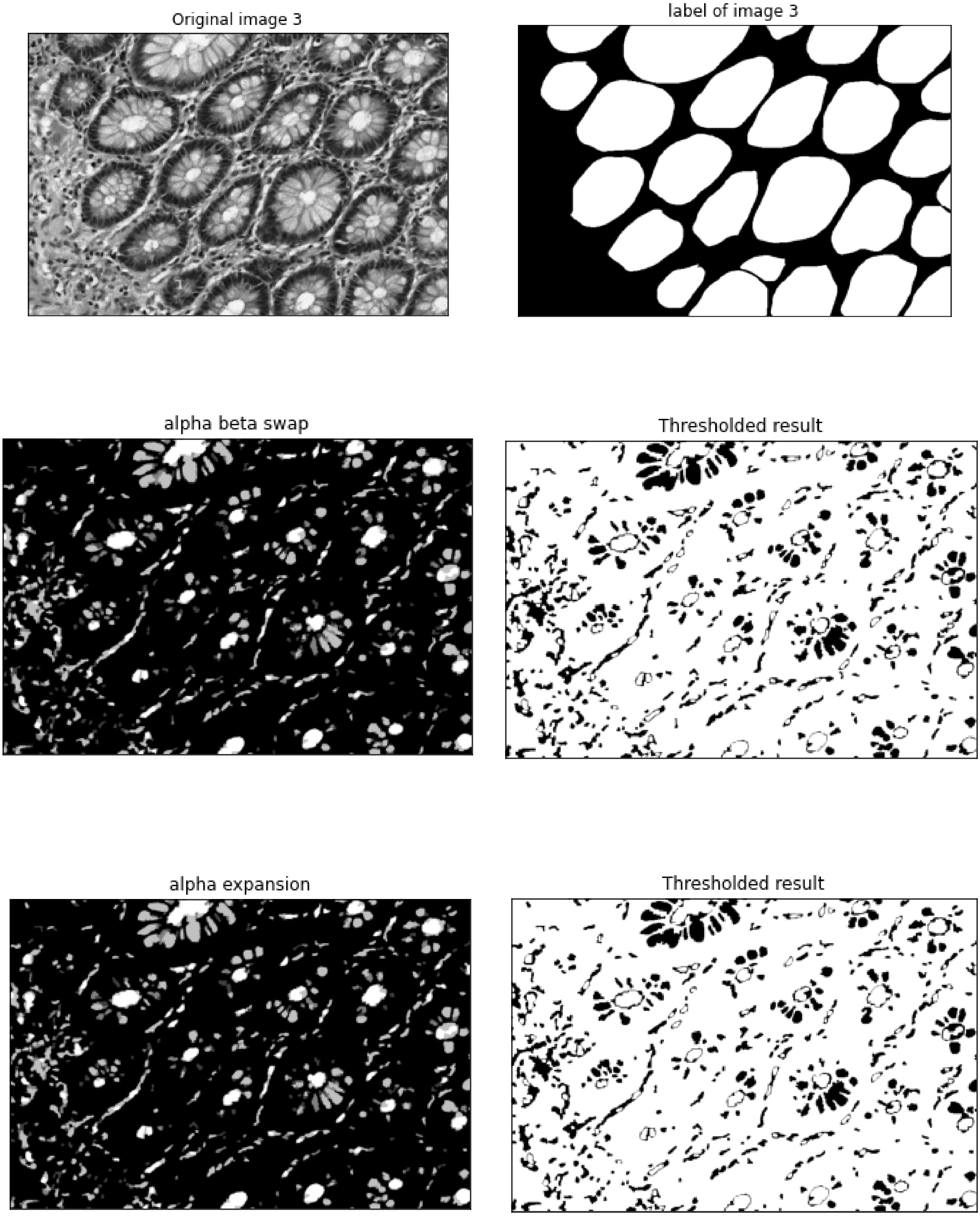
Results:



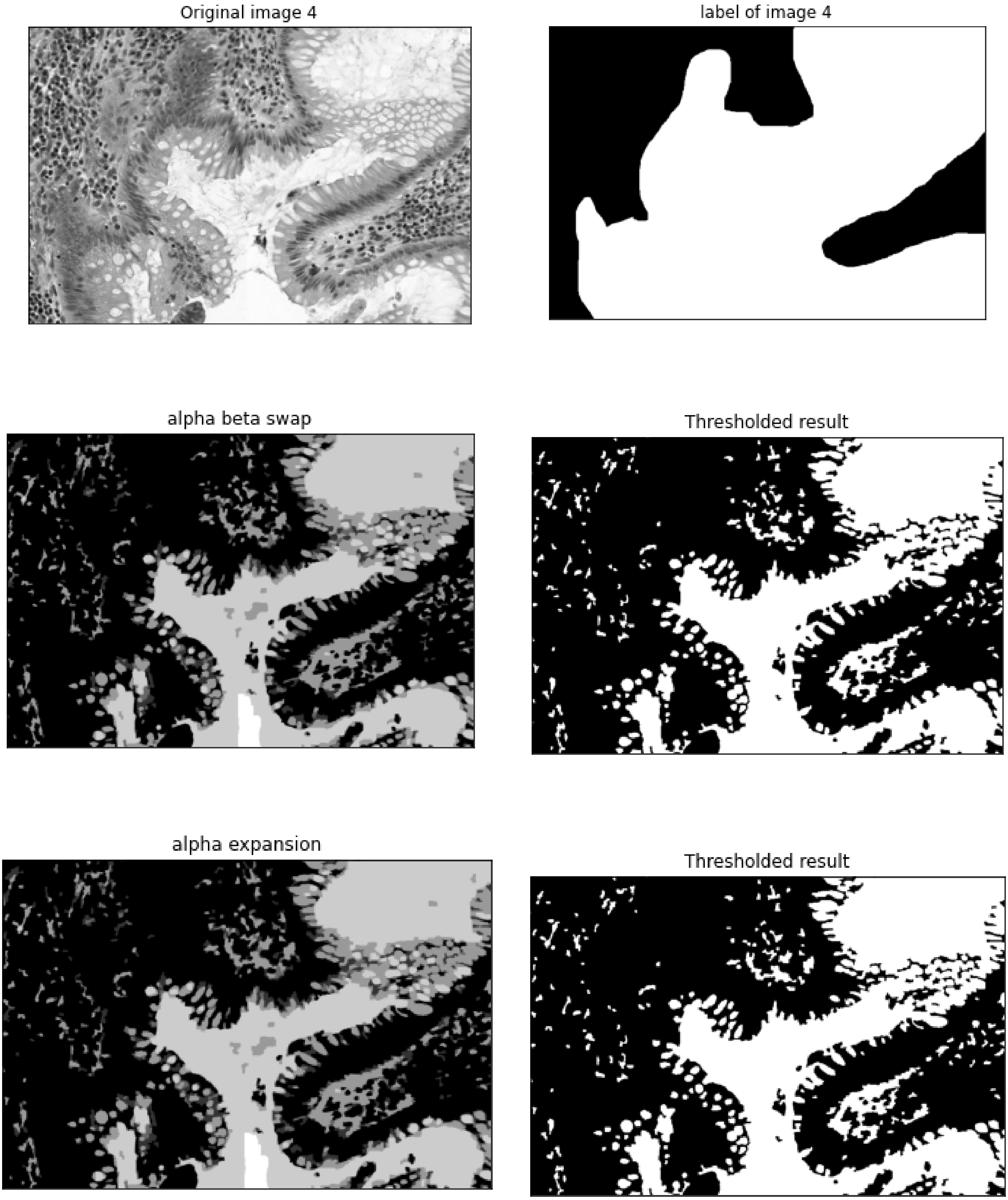
Results of image 1



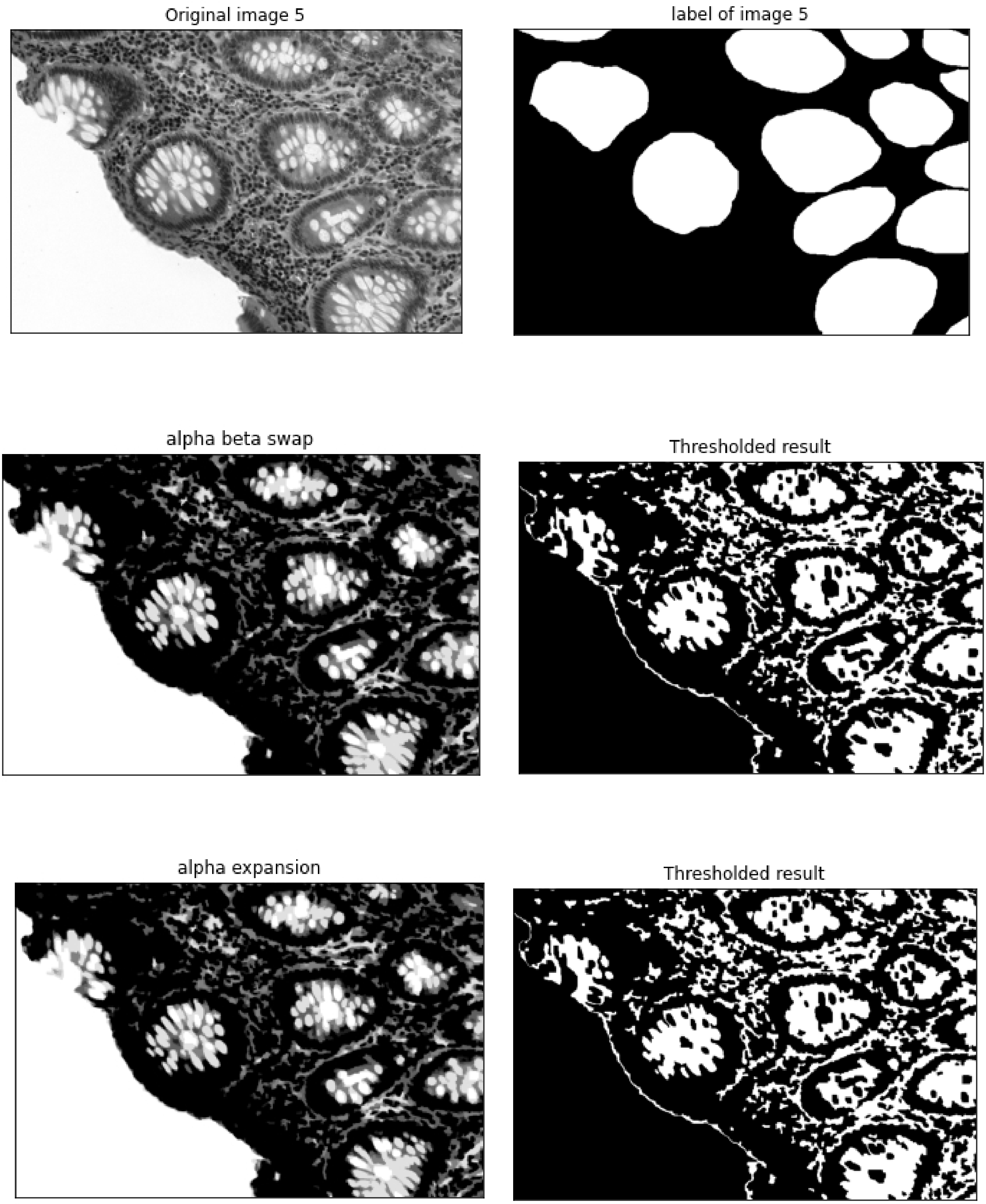
<u>Results of image 2</u>



Results of image 3



Results of image 4



<u>Results of image 5</u>

Observations:

There is only a very minute difference between alpha beta swap results and the alpha expansion method's results. The difference is observed only in a few pixels. The thresholded results, which were obtained by manually classifying the labels in the results of the alpha beta methods and the alpha expansion methods are not perfect and do not match the true labels perfectly.

3. Brief Description:

To compare the segmentation results with the ground truth using metrics like accuracy, Dice similarity coefficient, Jaccard index (JAC), sensitivity, specificity

Results:

alpha beta swap results:

	Dice	Accuracy	Sensitivity	Specificity	Jaccard index
Image 1	0.749	0.653	0.897	0.228	0.6
Image 2	0.698	0.581	0.814	0.162	0.536
Image 3	0.713	0.584	0.869	0.112	0.554
Image 4	0.645	0.63	0.503	0.441	0.476
Image 5	0.47	0.635	0.403	0.596	0.307

alpha expansion results:

	Dice	Accuracy	Sensitivity	Specificity	Jaccard index
Image 1	0.75	0.651	0.903	0.219	0.6
Image 2	0.7	0.58	0.82	0.154	0.537
Image 3	0.713	0.582	0.875	0.104	0.554
Image 4	0.644	0.63	0.5	0.44	0.475
Image 5	0.468	0.638	0.396	0.603	0.305