Irem Ural, 21502278

CS 484 – Image Anlaysis

HW 3 - Image Segmentation

1. Introduction

In this assignment, breast tumor tissues are analyzed by using feature extraction and image segmentation techniques by using Breast Cancer Histopathological Database [1]. Firstly, superpixel image segmentation technique is used by using SLIC algorithm[2], in order to divide image into small regions. Secondly, with the help of differently orientated and scaled Gabor filters, texture features are extracted for each pixel. Then, the average Gabor texture features are computed. To improve image segmentation, Lab color space is used and for each superpixel histogram is calculated. These two features are combined to have a better representation of each superpixel. K-means clustering is used for whole data set and the results can be seen in Part 7.1.Lastly, for each superpixel in the image, average neighbour features are calculated in two levels as first-level neighbourhood and second-level neighbourhood. As a result of this stage, K-means clustering algorithm is used again and the results can be find in Part 7.2 and comparisons of these two techniques can be find in the Discussion part.

2. Implementation Details

In the first part, slic zero algorithm is used, by setting parameter as 350, in order to create smaller subregions of the whole image [2]. In the second part, imgaborfilt() function is used with a grayscaled of each image and a filterbank for extracting texture features. In the third part lab color space is used to find color features of each superpixel. In this case, the l value is defined as between 0 and 100, while a and b values are defined as between -110 and 110. Each value is independently divided into 8 bins. For each superpixel, a histogram for l, a and b is created by counting the pixels in that superpixel. In LAB space, L defines the brightness factor, while a is used to detect colors between red and green axis and b is used to detect colors along the blue and yellow axis. Then these features are combined by normalization. Up to this point, all work is done in the following function in Matlab:

```
function [output1, output2] = findRepresentation(fileName)
```

It gets filename as a parameter reads the image and applies the above techniques to get concatenated features for each superpixel of the image.

In part 4, Matlab's K-Means clustering algorithm is used. Clustering is applied to a matris which contains all superpixels of all 10 images. Then false color image representation is acquired by finding each pixel's clusterID and creating a label matrix which has clusterIDs. It is defined by each image. Afterwards, by using imfuse() function of Matlab with blend as a parameter,the results of clustering is demonstrated on top of the image.

In the last part of the assignment, by using the label matrix achieved from slic 0 algorithm, for each image regionprops() function is called in order to get the center of superpixels and their radius. The radius is calculated according to a circle with same area [3]. Then a matrix is created with the same dimensions as the image. The ones, that are in the circle are set as 1, it is achieved by using following two lines of code:

```
[rows,columns] = meshgrid(1: columnImage, 1:rowImage);
 pixels = (rows-centerX).^2 +(columns-centerY).^2 <= radius.^2; [4]
```

For first neighbourhood, radius is multiplied by 2 and for the second neighbourhood radius is multiplied by 3. For first neighbourhood labeled 1's are created by subtracting the 'pixels' defined above. For the second neighbourhood, both first neighbourhood and 'pixels' are subtracted from second neighbourhood. As a result of this the locations of the pixels that belong to a neighbourhood is obtained. By using these two matrices, the superpixelID's are obtained according to the label obtained from slic 0 algorithm. Then the average features are calculated for each neighbourhood. Furthermore, a threshold is used to accept superpixels as neighbours, it is set to 50 in the implementation. As a result of this part, a matrix of N(number of superpixels) x 120 features is achieved. Then clustering algorithm is used again on the created matrix and false color representations of each image is demonstrated.

How to run the code?

The image files should be in the same directory with the matlab file. The file can be executed by the run button of the Matlab or by typing findsuperpixels.

3. Description of parameters used for superpixel segmentation

There are two algorithms provided. One of them is slic zero and the other one is slic algorithm.

Slic0 algorithm expects k as the parameter, which is the desired number of equally sized superpixels. It automatically adapts the m value (which is used to arrange the contents of the superpixels in texture and non-texture areas).

Slic algorithm expects k and m values. In this case user chooses the compactness factor, as m increases more compact superpixels are formed. When m is small, the superpixels are adheres more to the image boundaries and have less regular size and shape [1].

Slic0 algorithm is used in the first part. It When k increases number of superpixels increase.

I first tried the k = 100 on the first image.

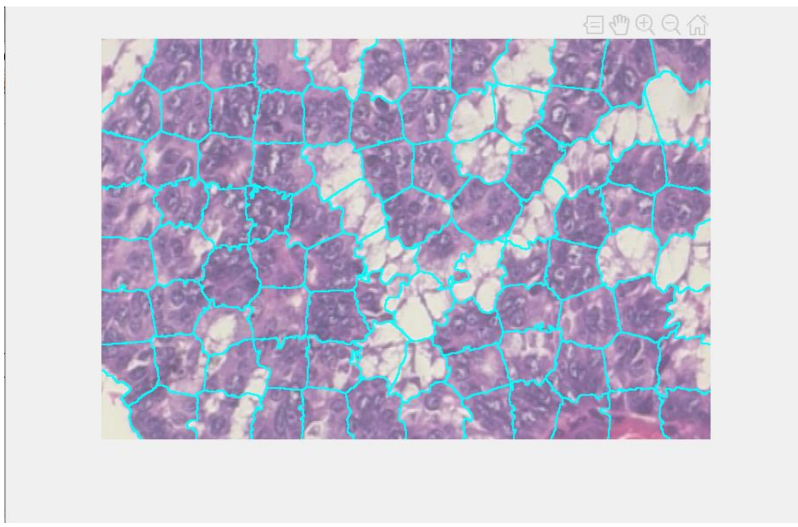

Figure 1. Image 1 Superpixel result (number of superpixels is set to 100)

The cyan boundaries shows the superpixels that are generated. Then I incremented the value to 250. The result better suited as it contains a few pixels on the image. After analyzing all the images select number of superpixels as 350.

Compared to the results of slic algorithm, slic0 provides more compact and rigid superpixels.

4. <u>Segmentation Result for All Images Using Slic Zero Algorithm</u>
<u>Number of Superpixels = 350</u>

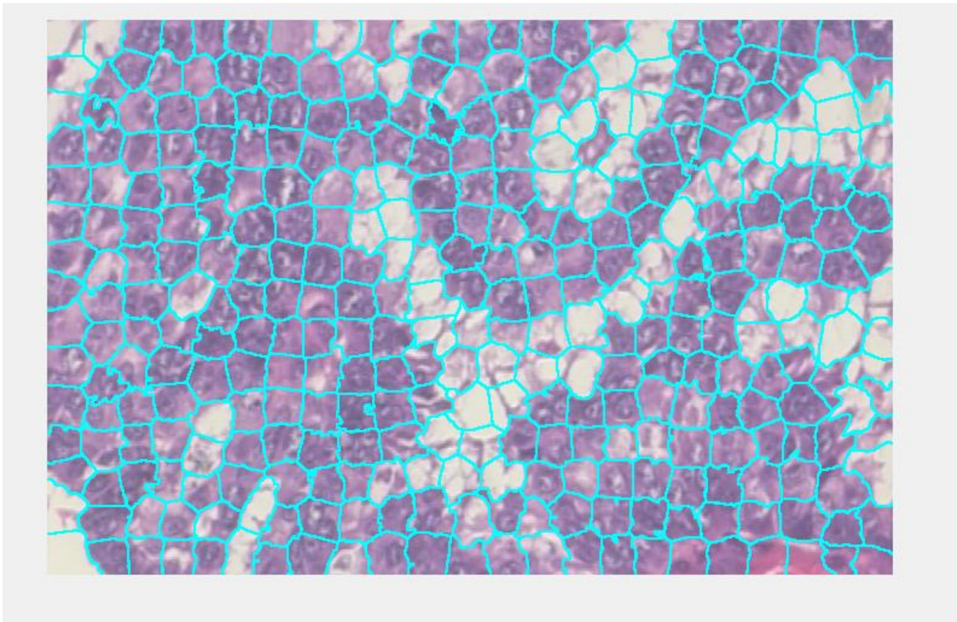

Figure 2. Image 1 Segmentation Result

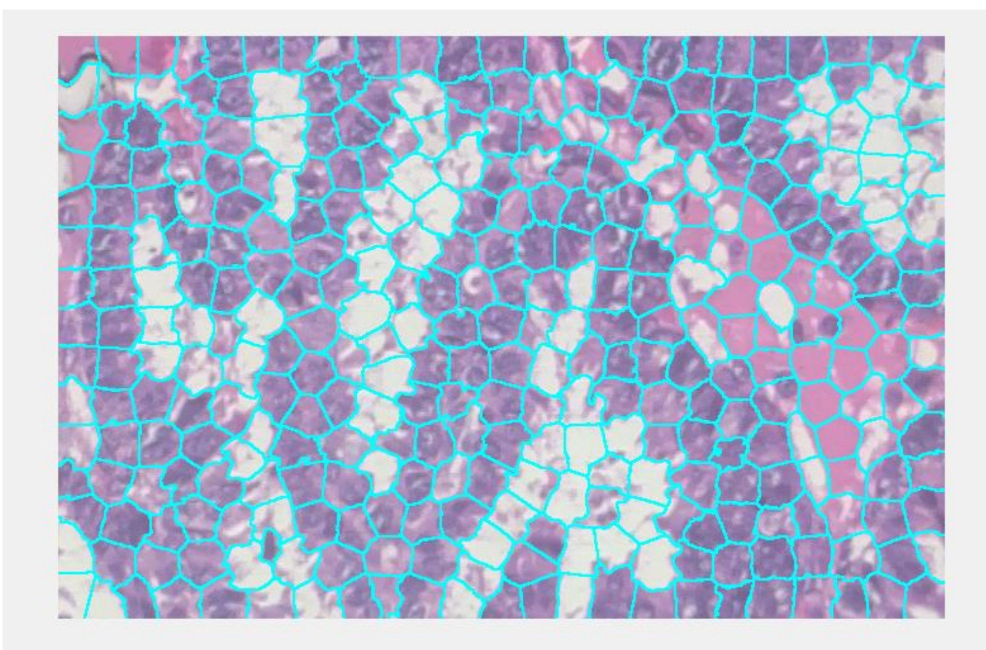

Figure 3. Image 2 Segmentation Result

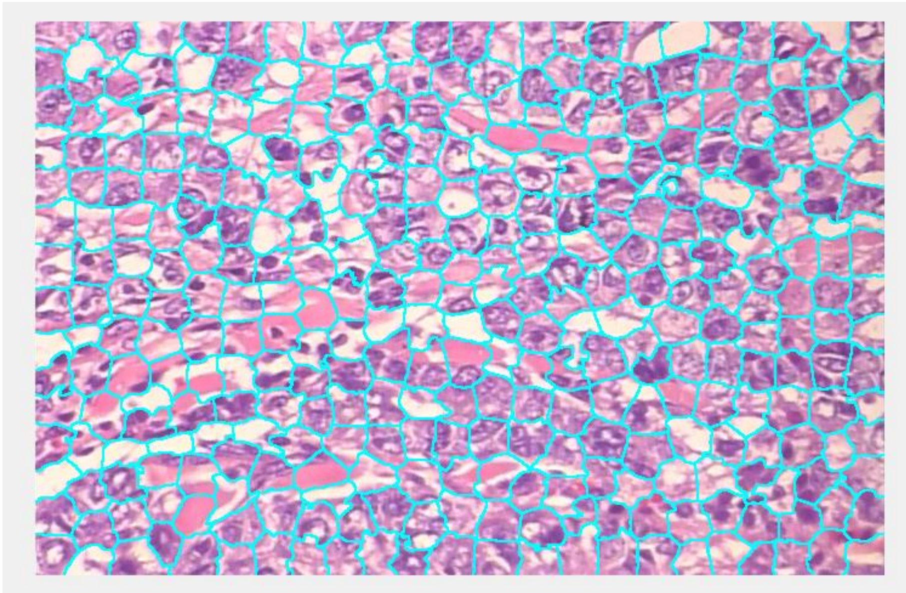

Figure 4. Image 3 Segmentation Result

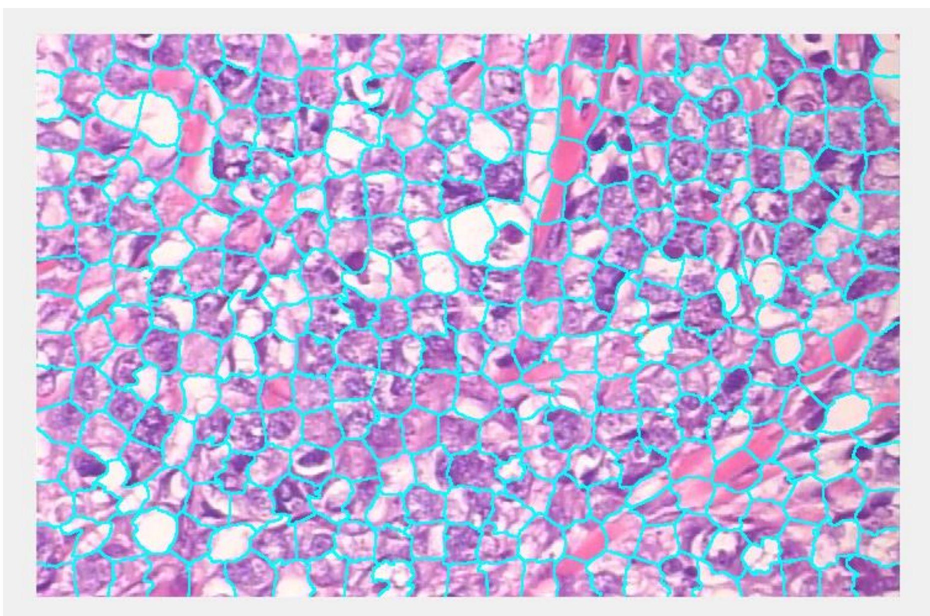

Figure 5. Image 4 Segmentation Result

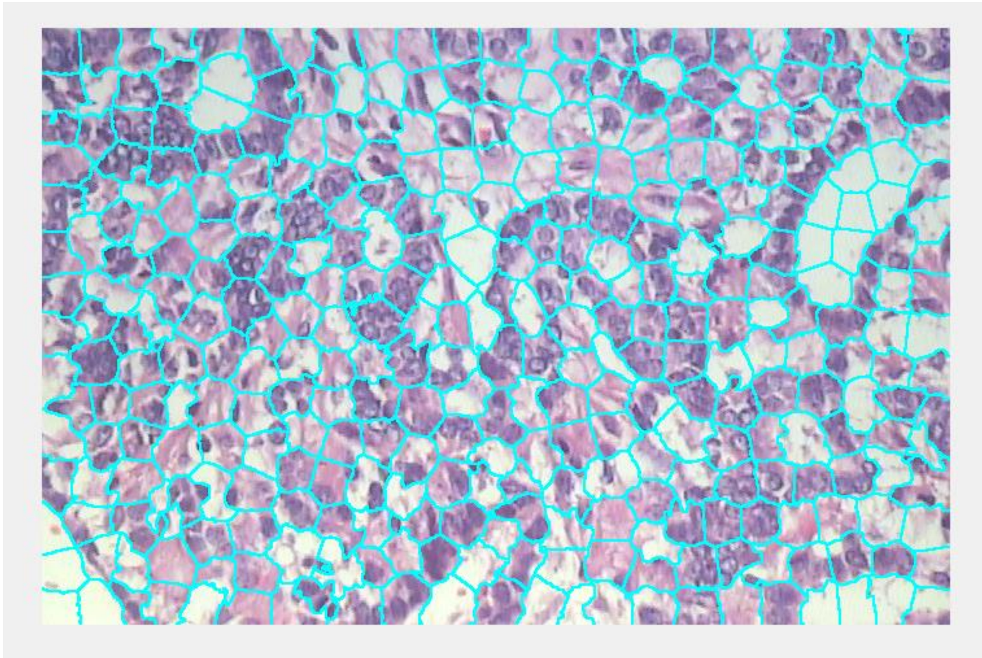

Figure 6. Image 5 Segmentation Result

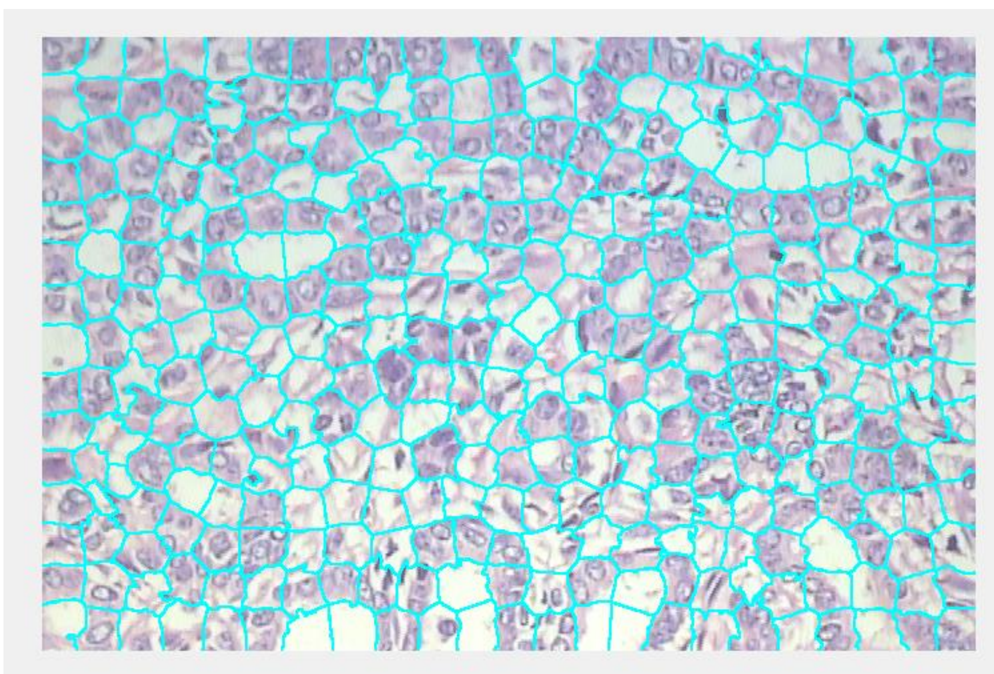

Figure 7. Image 6 Segmentation Result

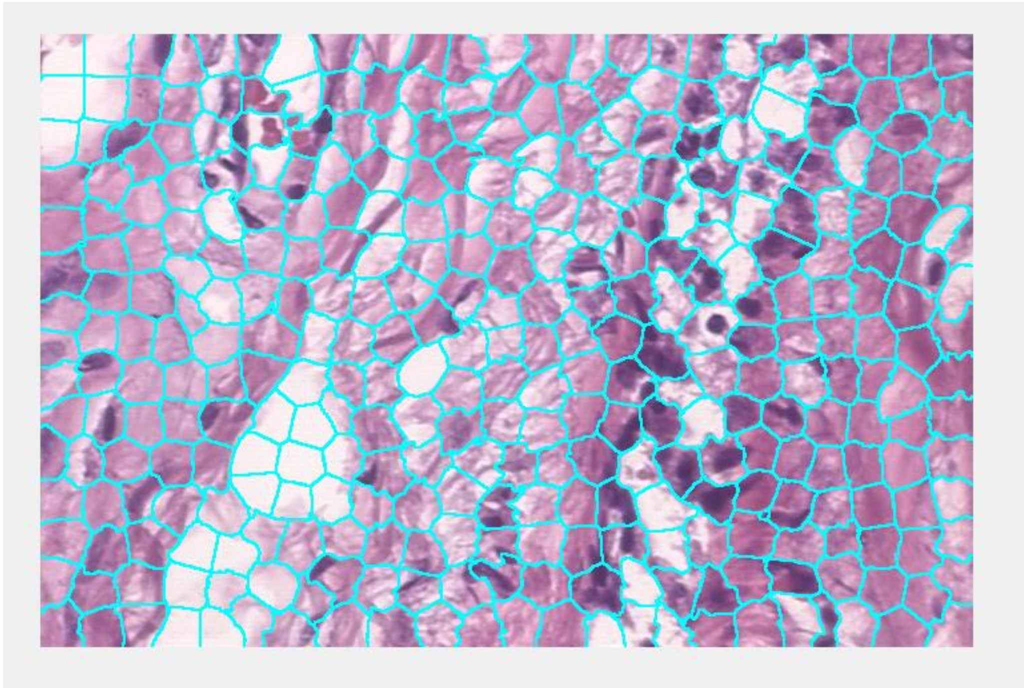

Figure 8. Image 7 Segmentation Result

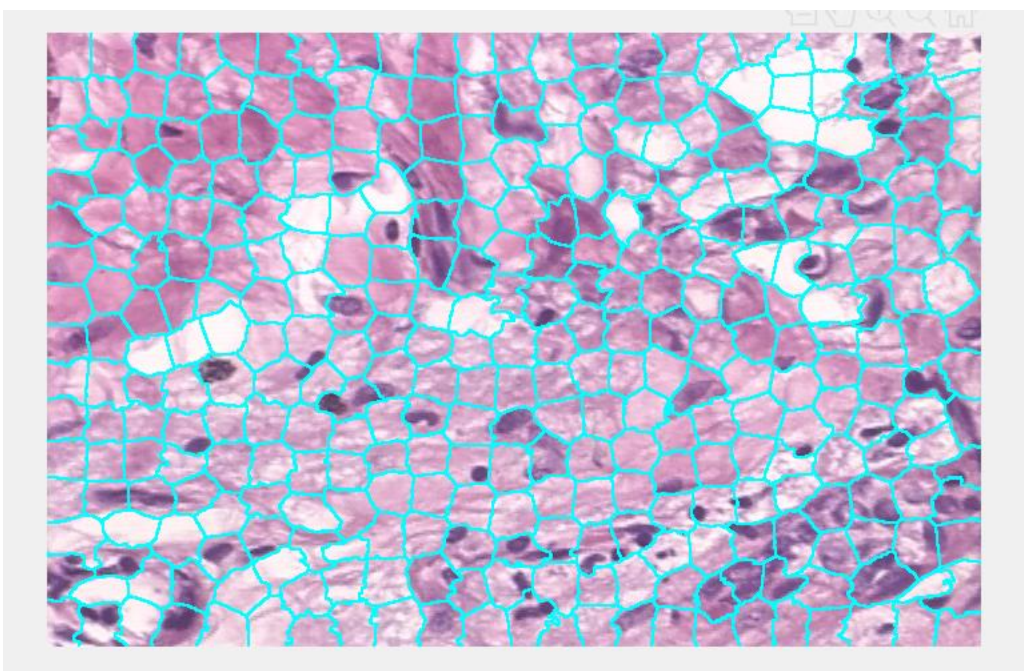

Figure 9. Image 8 Segmentation Result

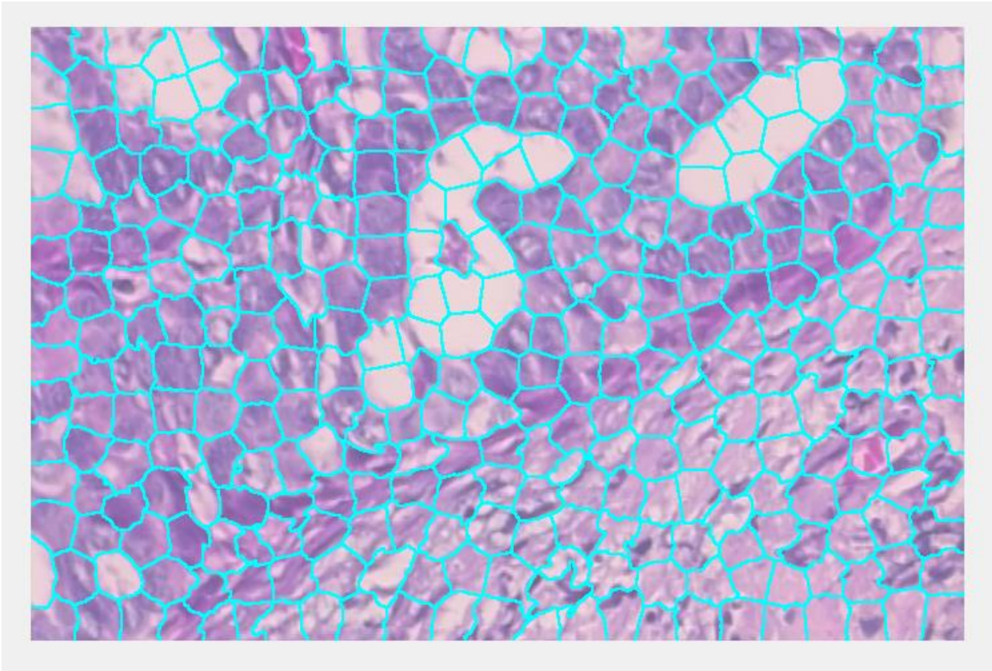

Figure 10. Image 9 Segmentation Result

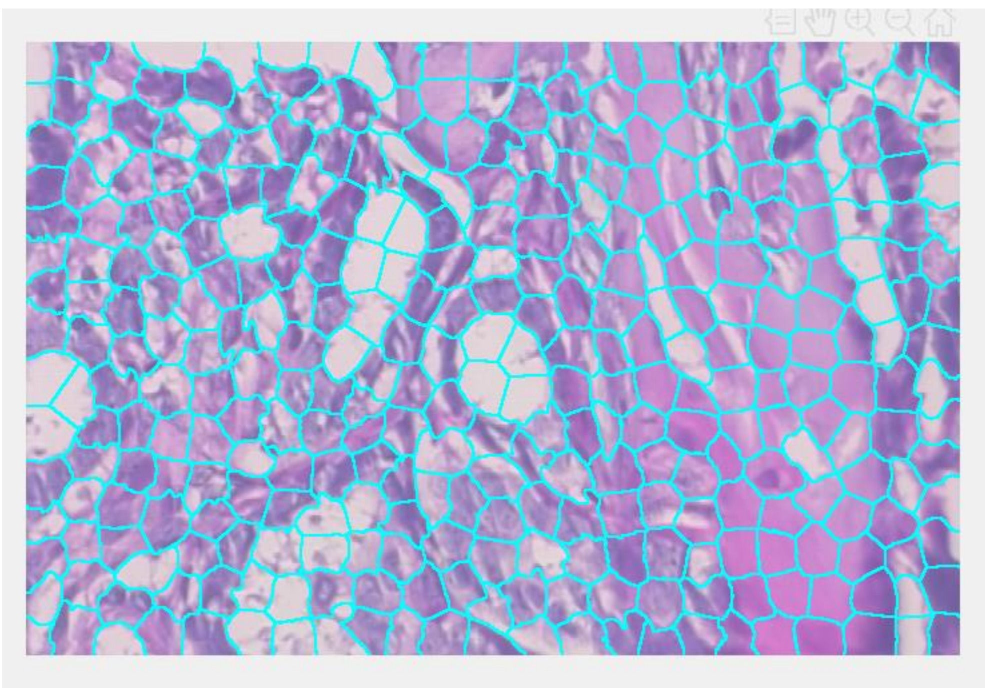

Figure 11. Image 10 Segmentation Result

5. Description of parameters used for Gabor texture feature extraction

Imgaborfilt function is used in the homework. It computes the magnitude and phase responses of grayscaled images to Gabor filters. It gets wavelength and orientation as parameter.

Parameter 1 :Gray-scaled image is expected as a parameter to compute its response to gabor filter.

Parameter 2 :Wavelength is the wavelength pixels/ cycle for sinusoidal carrier. If wavelength increases it responds frequency of the filter decreases, therefore it does not detect small changes in the image.

Parameter 3 : Orientation is expected as a parameter, it is the orientation of the parameters in degrees. According to the orientations, it detects changes in those directions.

6. Gabor Texture Feature Examples

Image 1:

Wavelengths:[10 20 30 40],
Orientations: [0 45 90 135]

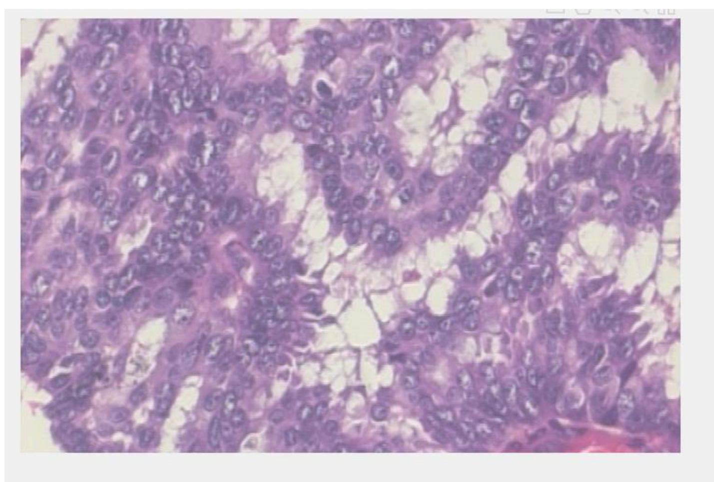

Figure 11. Image 1

Gabor filter response of the image 1:

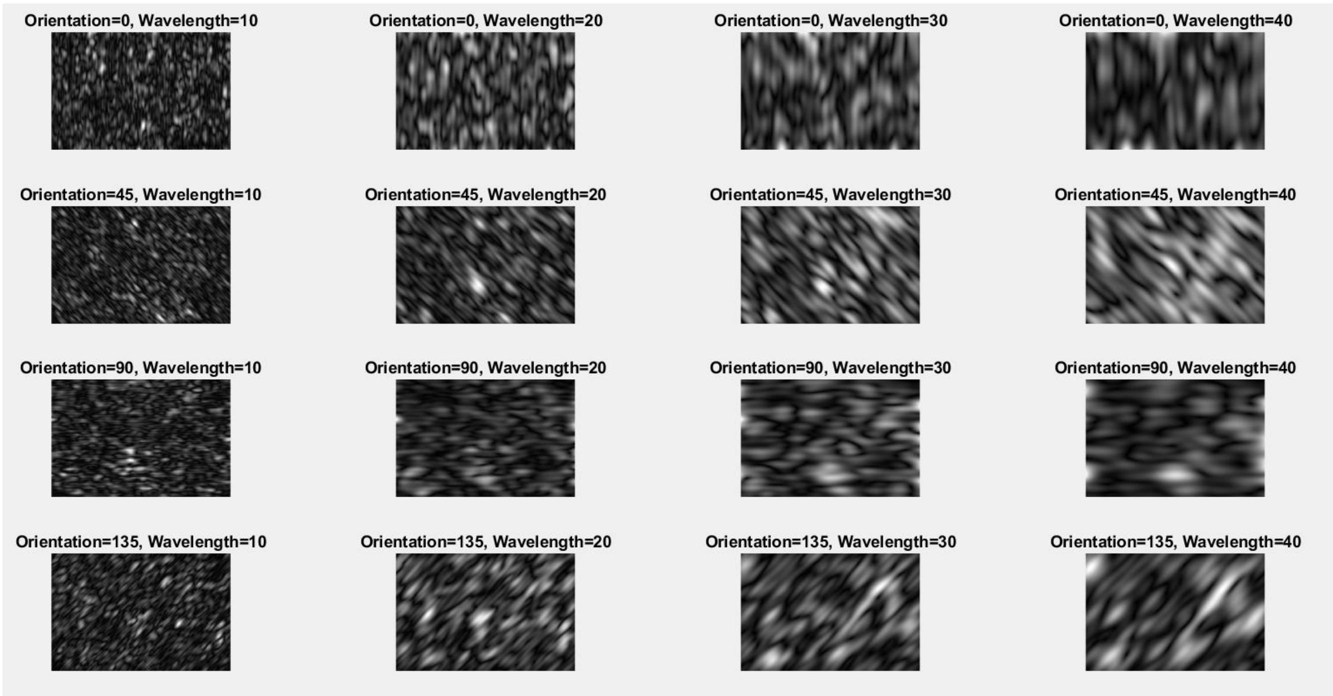

Figure 12. Image 1- Gabor Filter Result

Image 2:

Wavelengths:[10 20 30 40],
Orientations: [0 45 90 135]

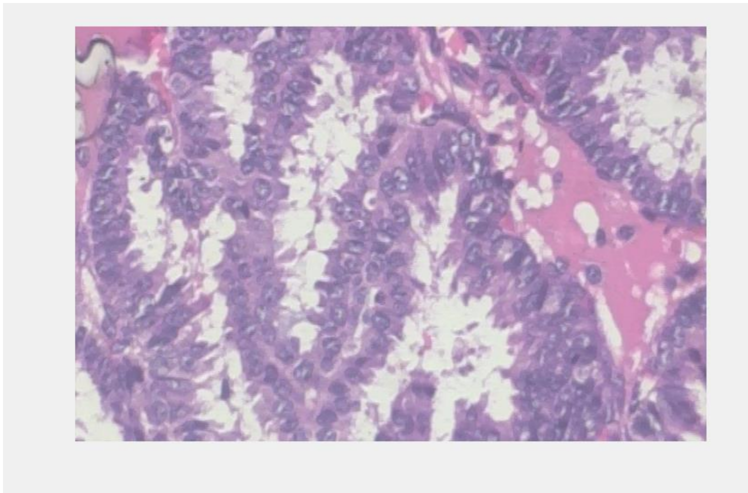

Figure 13. Image 2

Gabor filter response of the image 2:

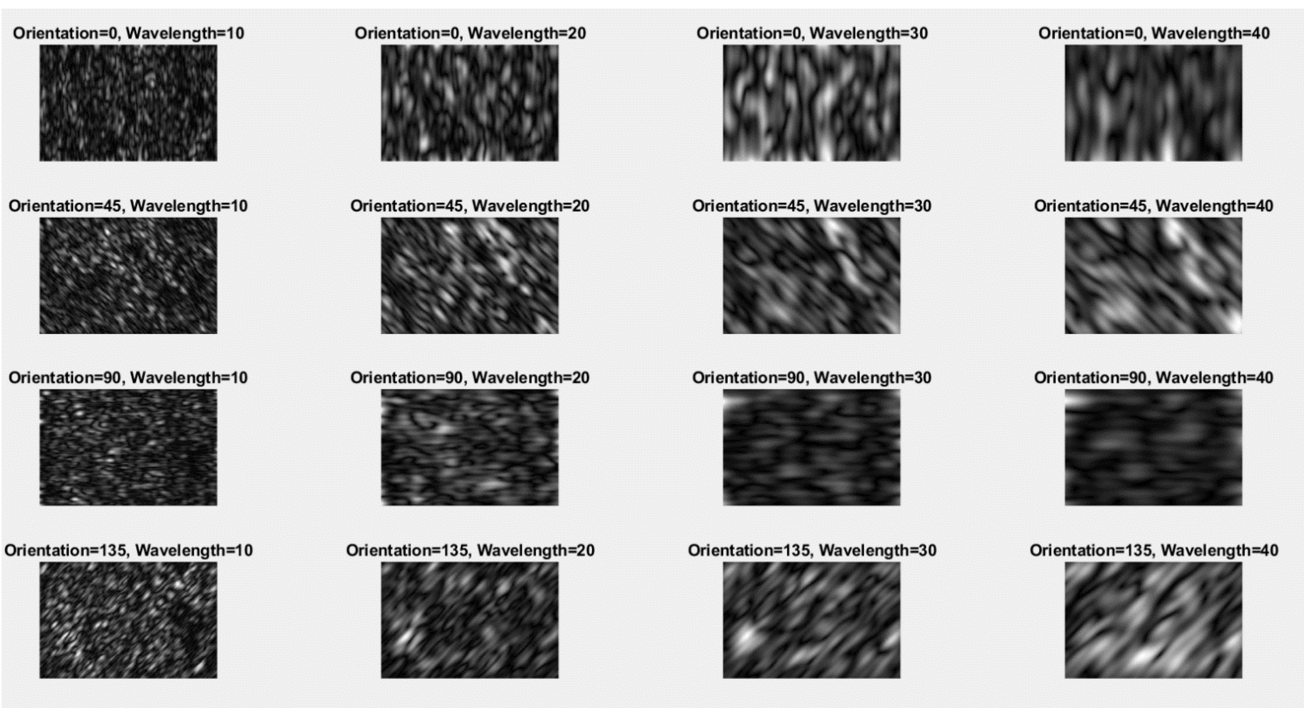

Figure 14. Image 2- Gabor Filter Result

<u>Image 3:</u>

wavelengths = [2 4 6 8]
orientations = [0 45 90 135]

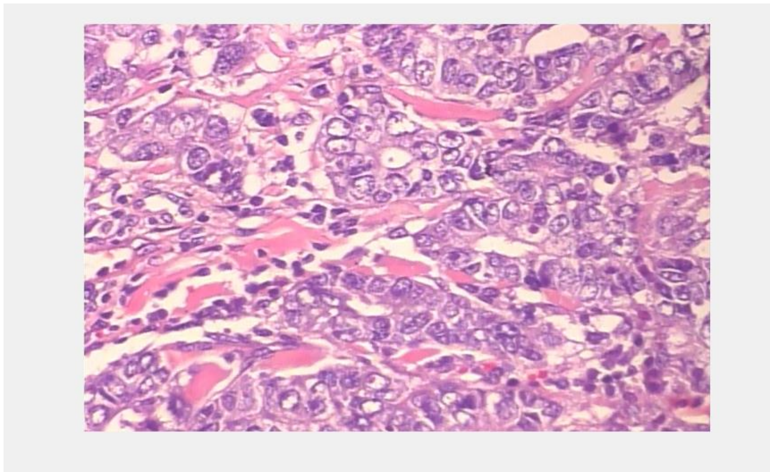

Figure 15. Image 3

<u>Gabor filter response of the image 3:</u>

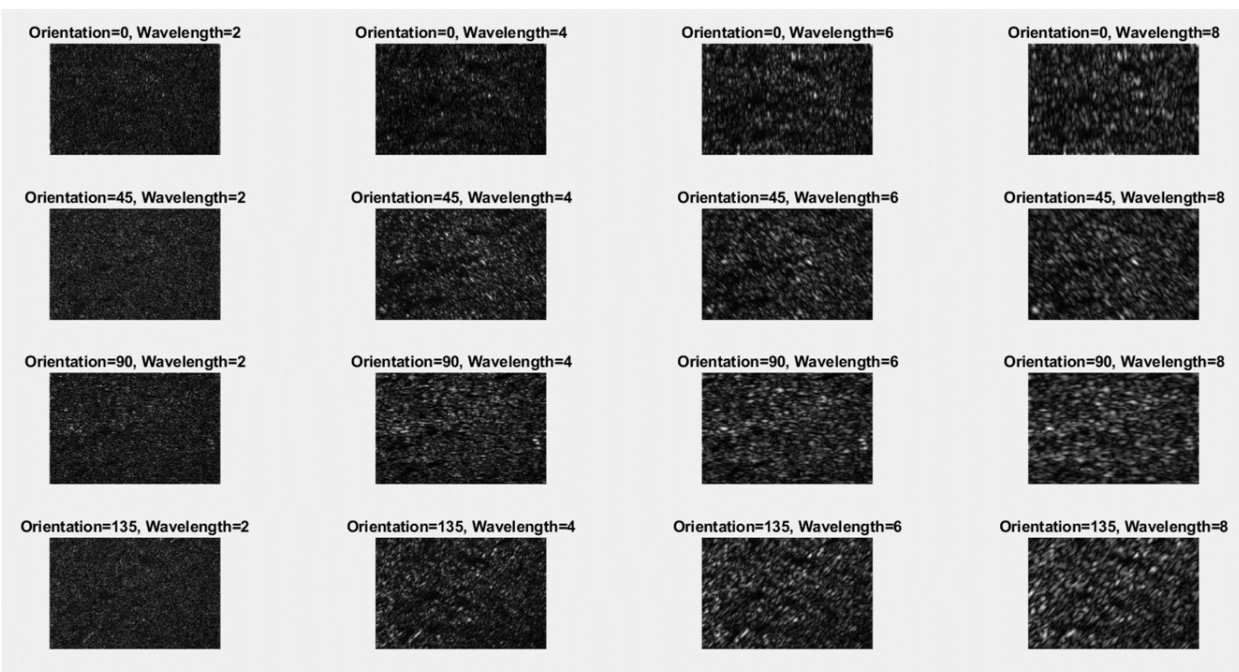

Figure 16. Image 3- Gabor Filter Result

Image 4:

wavelengths = [10 20 40 80]
orientations = [0 45 90 135]

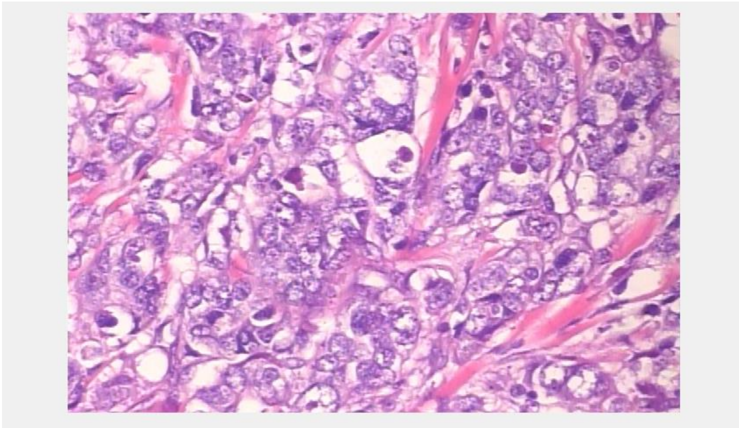

Figure 17. Image 4

Gabor filter response of the image 4:

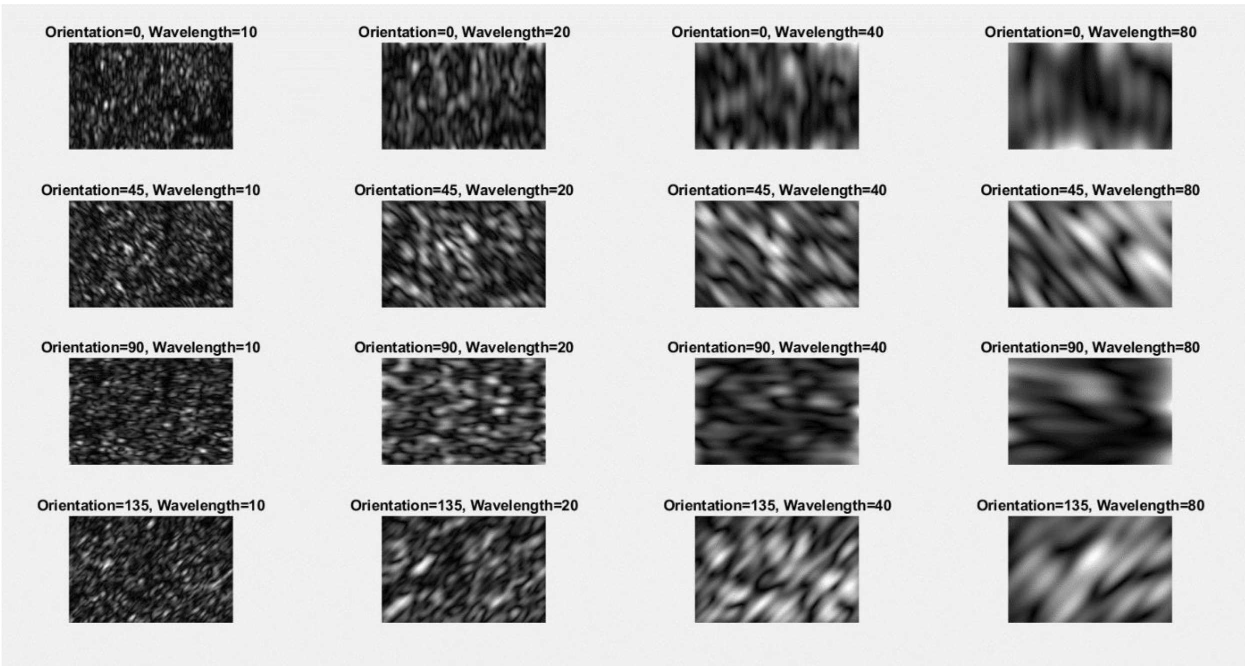

Figure 18. Image 4- Gabor Filter Result

Image 5:
wavelengths = [5 10 15 20]
orientations = [0 45 90 135]

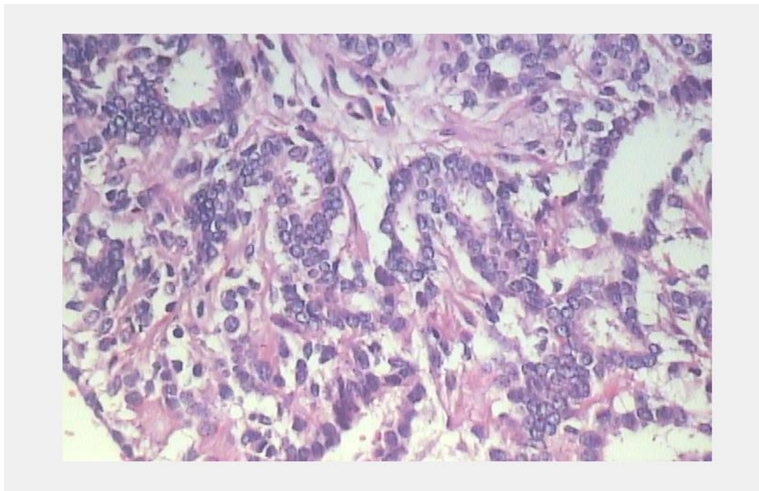

Figure 19. Image 5

Gabor filter response of the image 5:

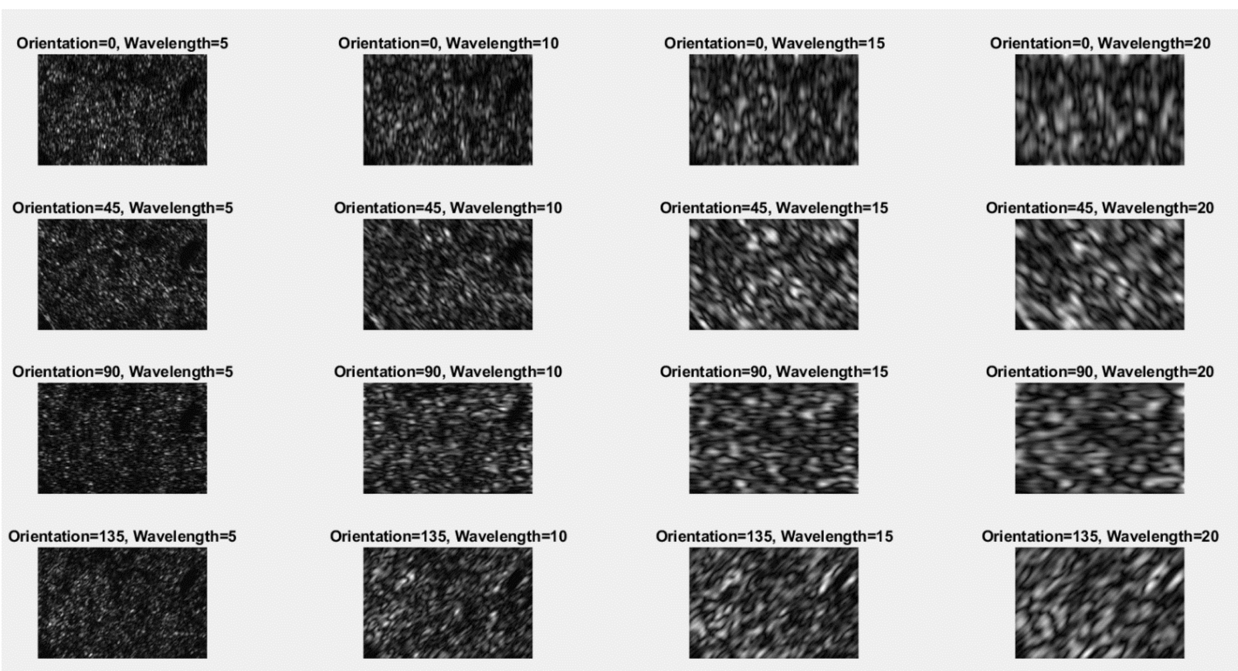

Figure 20. Image 5- Gabor Filter Result

7. Clustering and Segmentation Results

7.1 Part 4 Results

Number of Clusters: 350
K-means Clustering Parameter: 20
Gabor Filter Scales: [5 10 15 20]
Gabor Filter Orientations: [0 45 90 135]

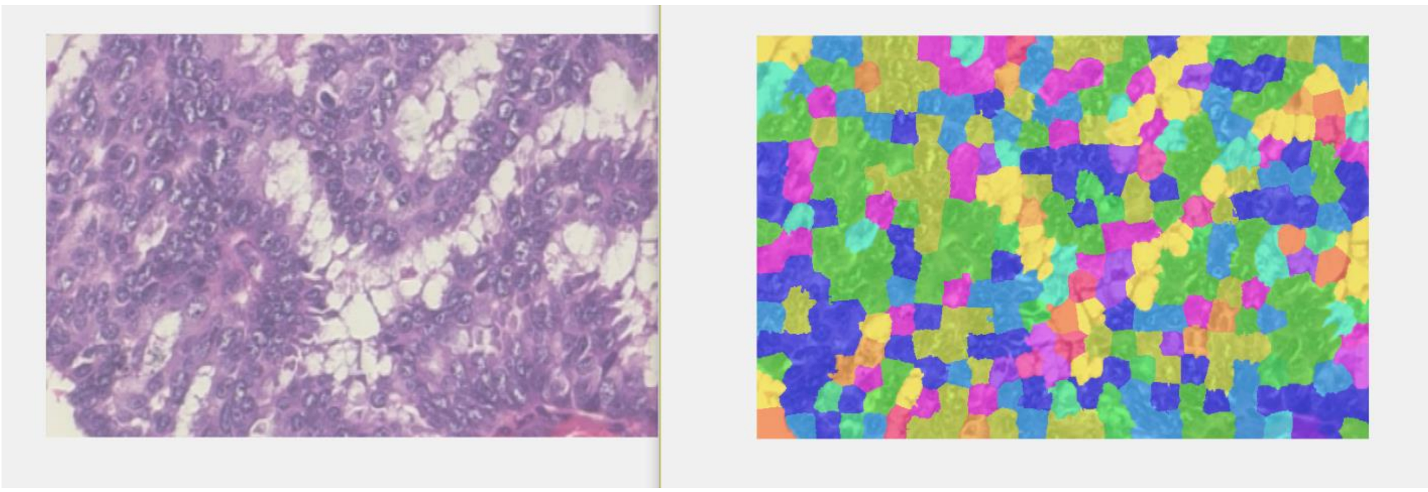

Figure 21. Image 1- Part 4 Result

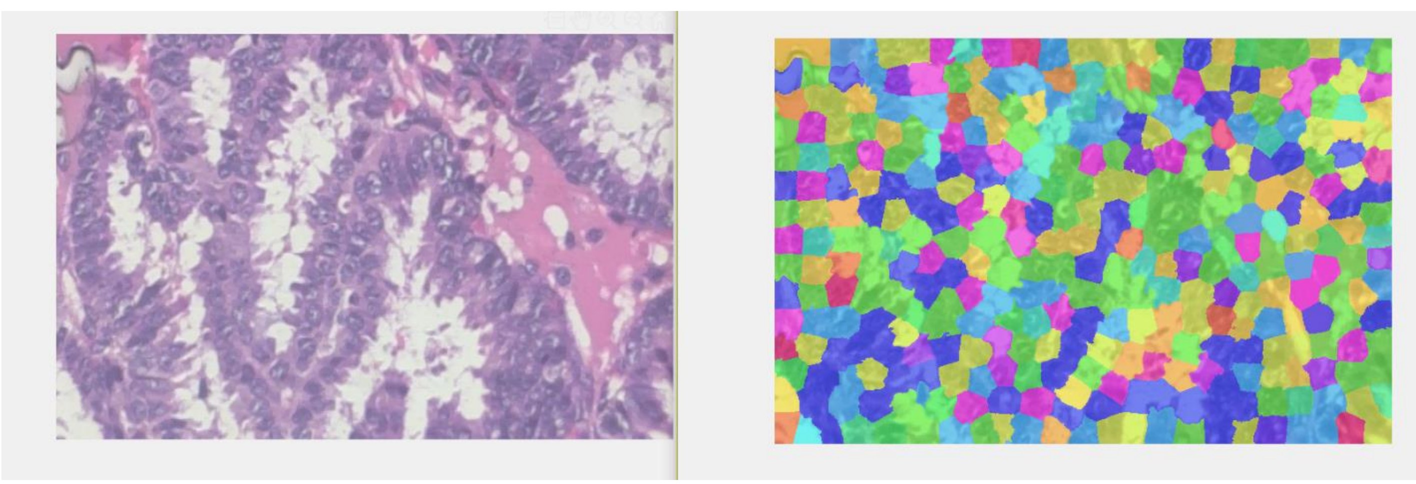

Figure 22. Image 2- Part 4 Result

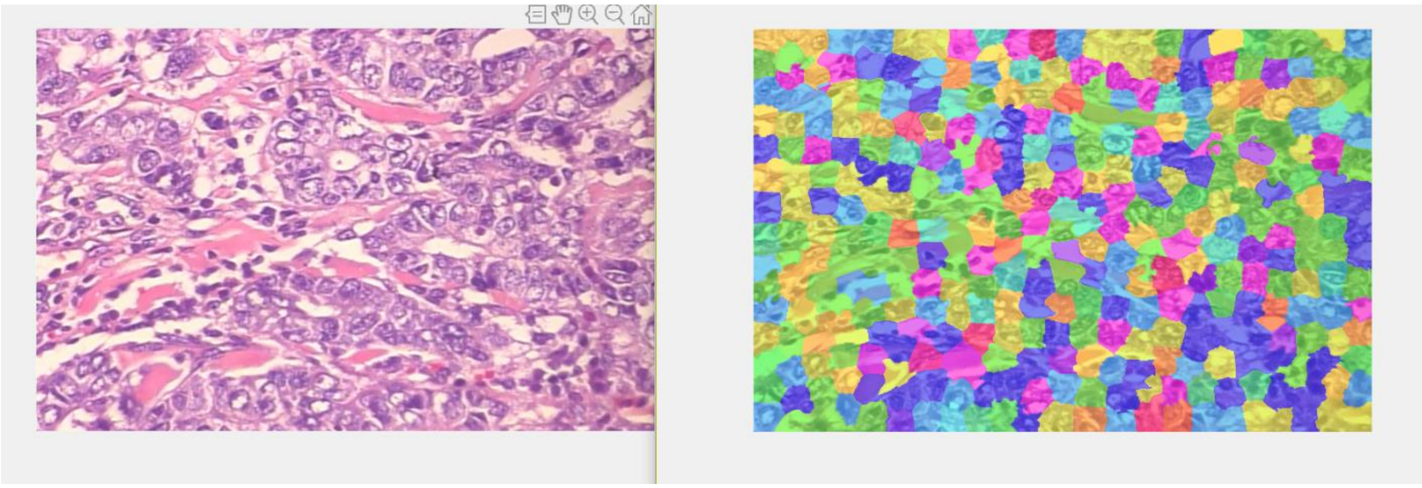

Figure 23. Image 3- Part 4 Result

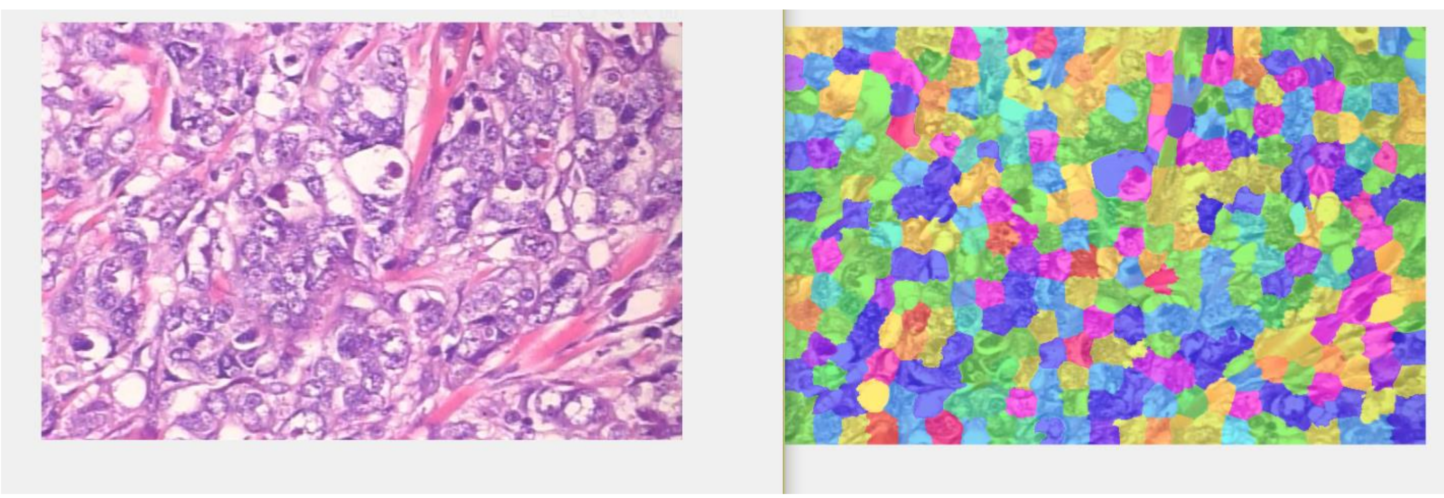

Figure 24. Image 4- Part 4 Result

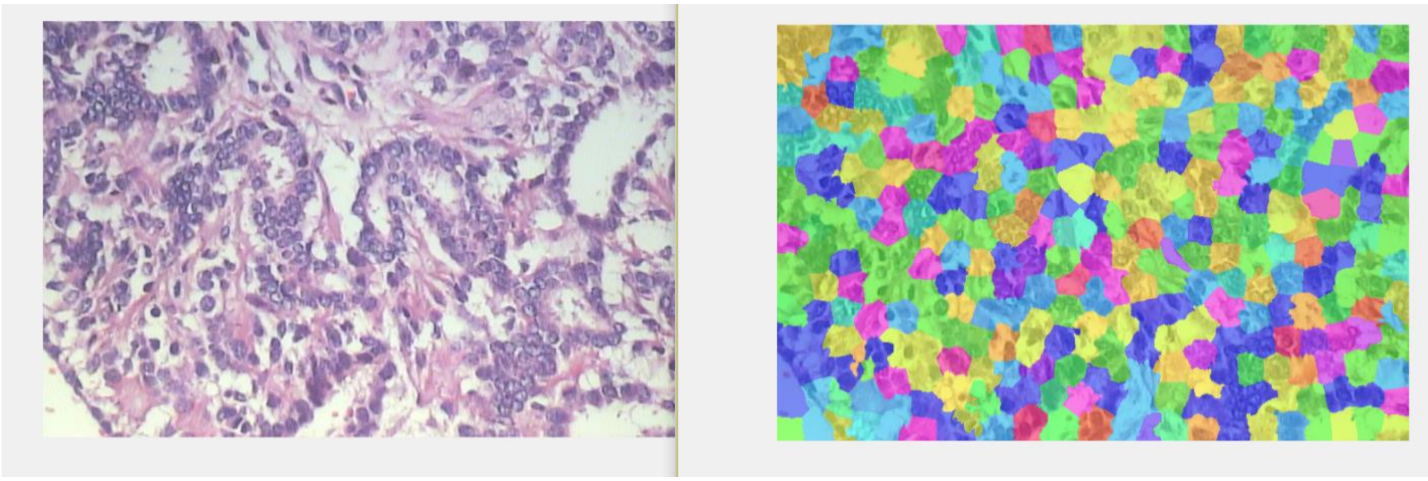

Figure 25. Image 5- Part 4 Result

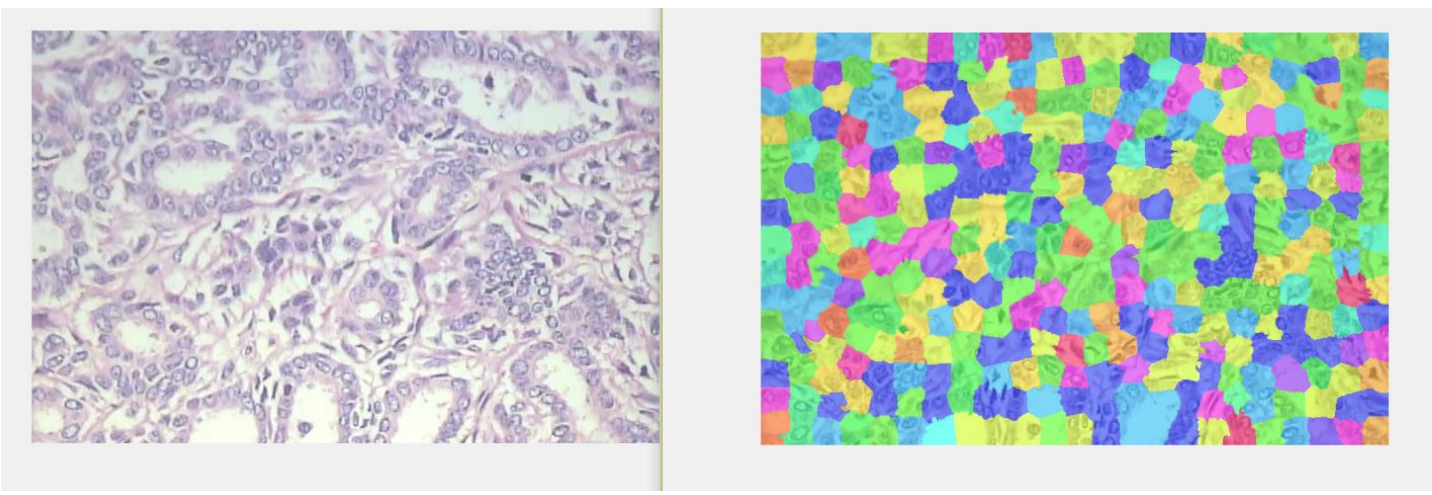

Figure 26. Image 6- Part 4 Result

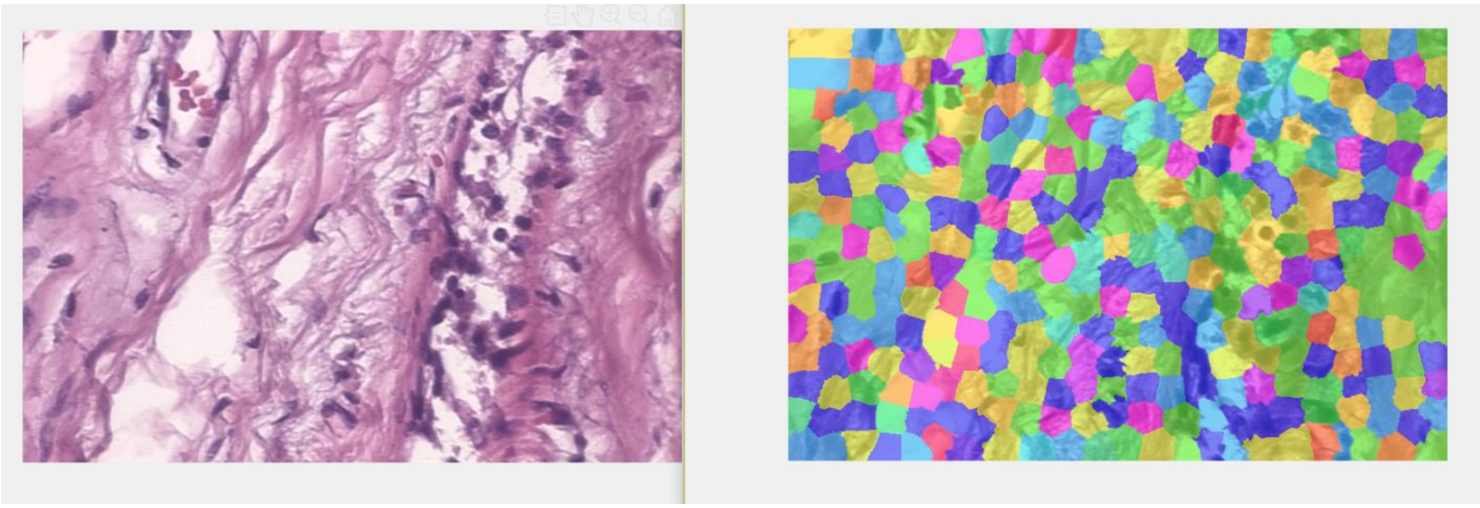

Figure 27. Image 7- Part 4 Result

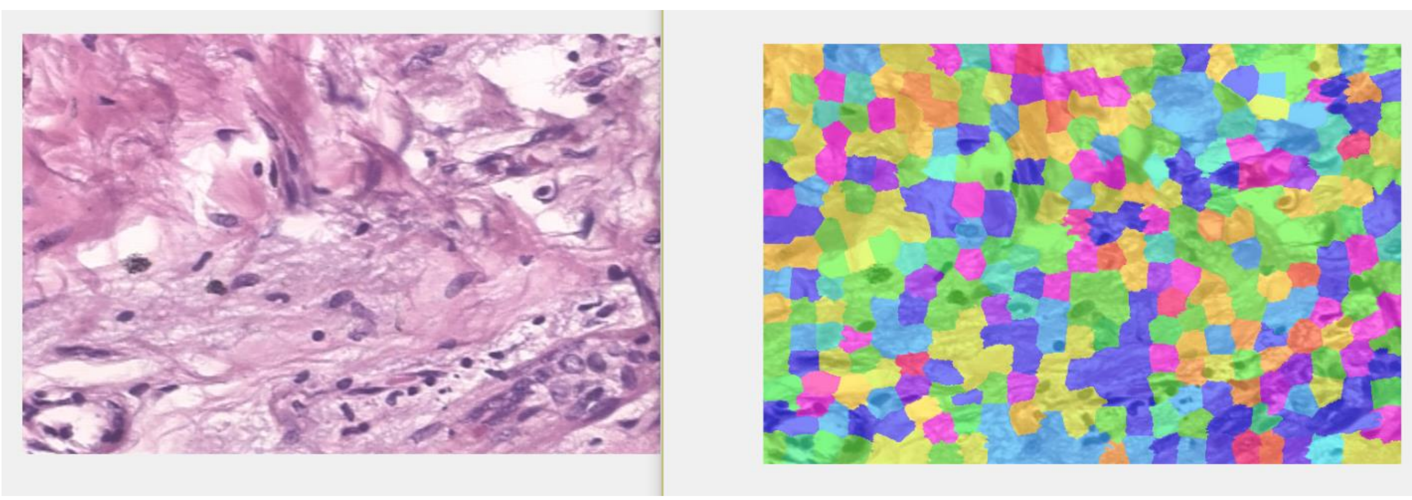

Figure 28. Image 8- Part 4 Result

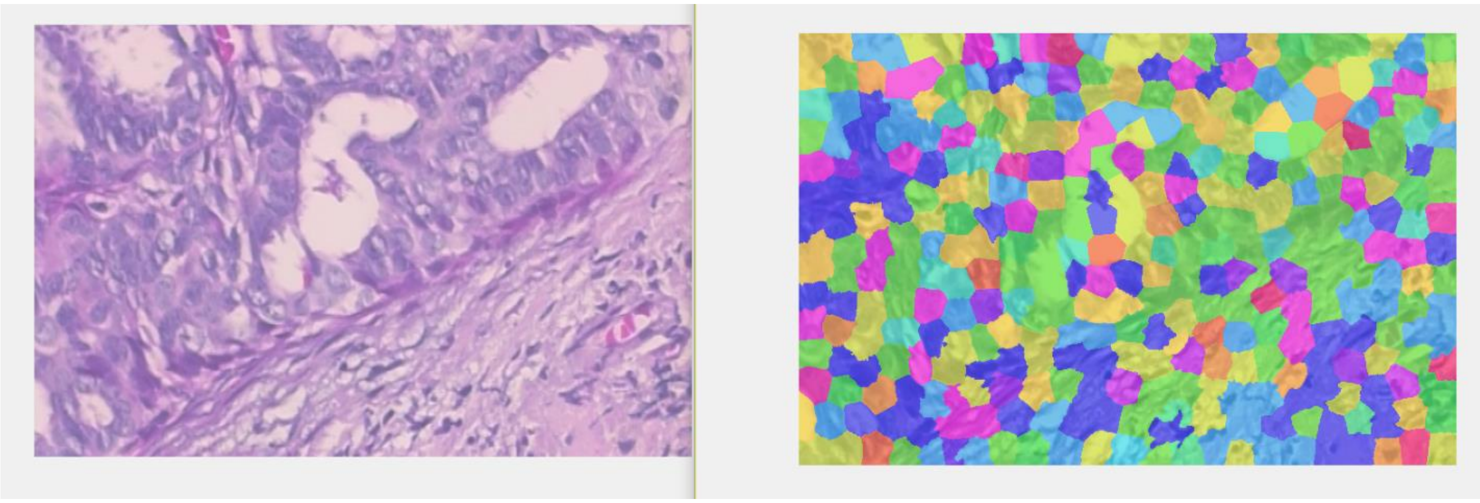

Figure 29. Image 9- Part 4 Result

Figure 30. Image 10- Part 4 Result

7.2 Part 5 Results

Threshold: 50

R1: 2

R2: 3

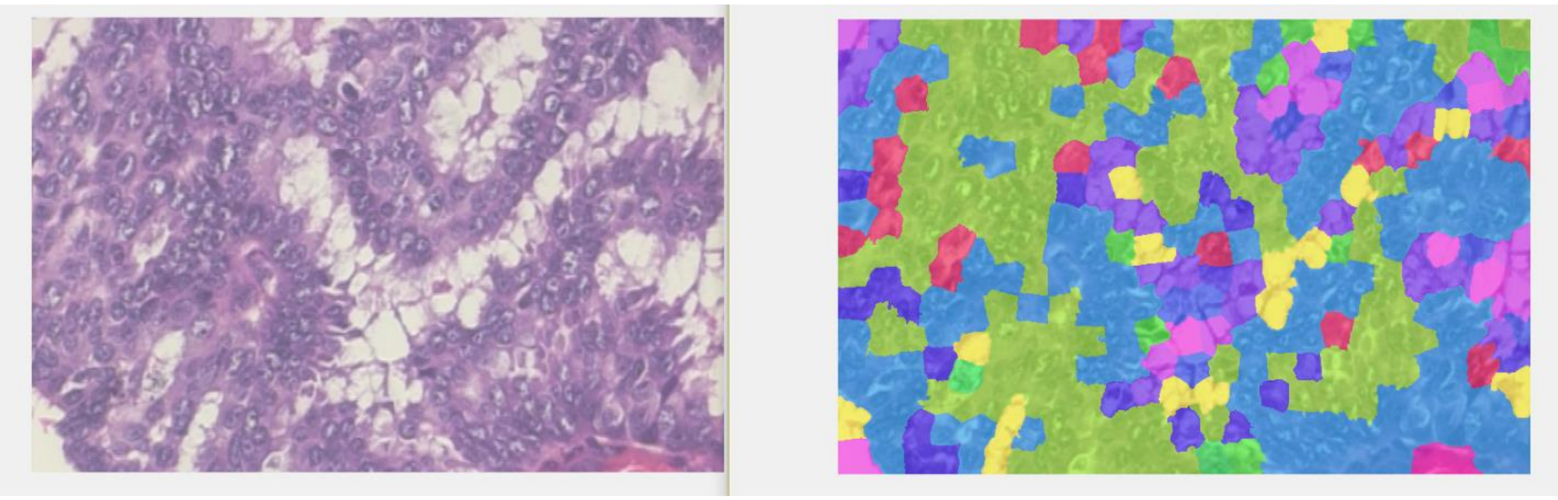

Figure 31. Image 1- Part 5 Result

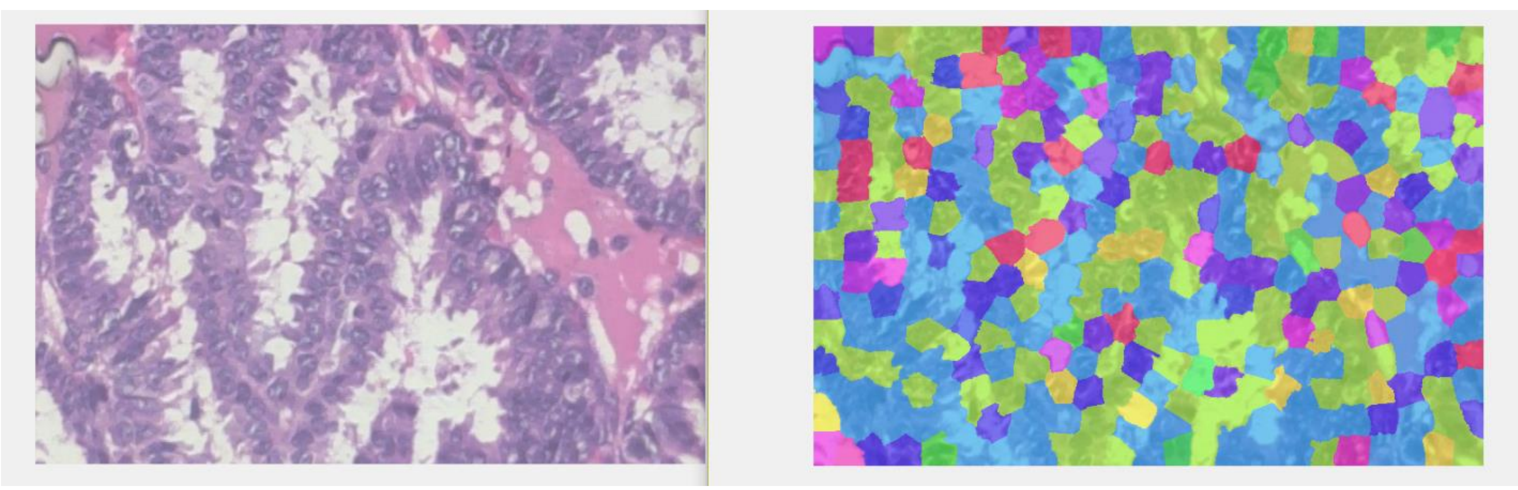

Figure 32. Image 2- Part 5 Result

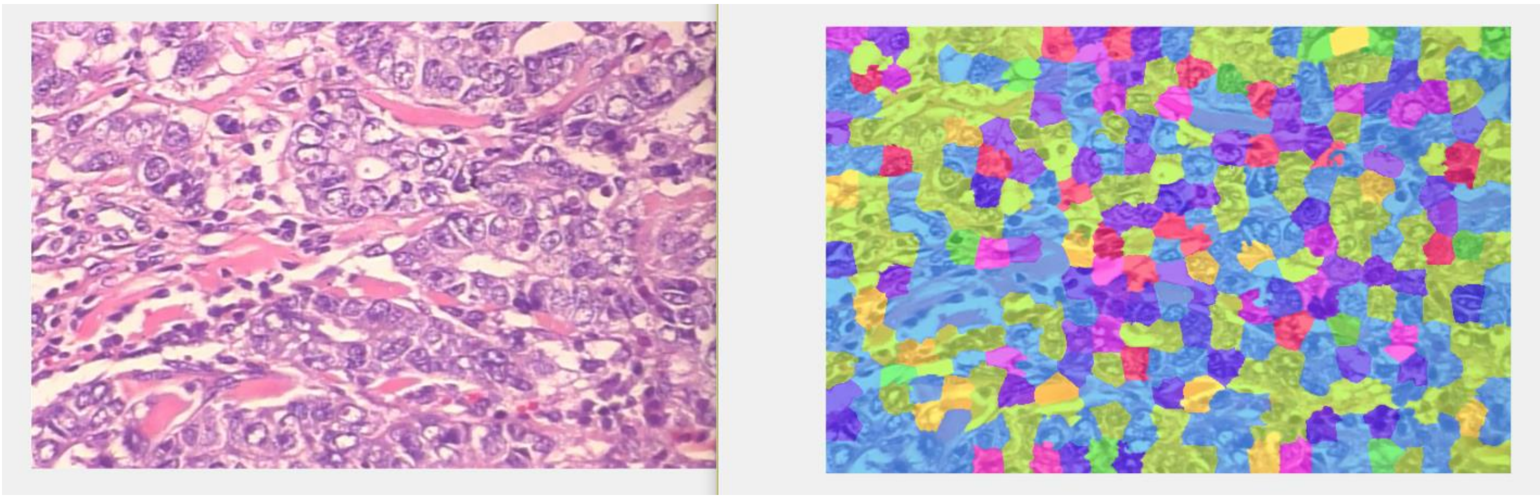

Figure 33. Image 3- Part 5 Result

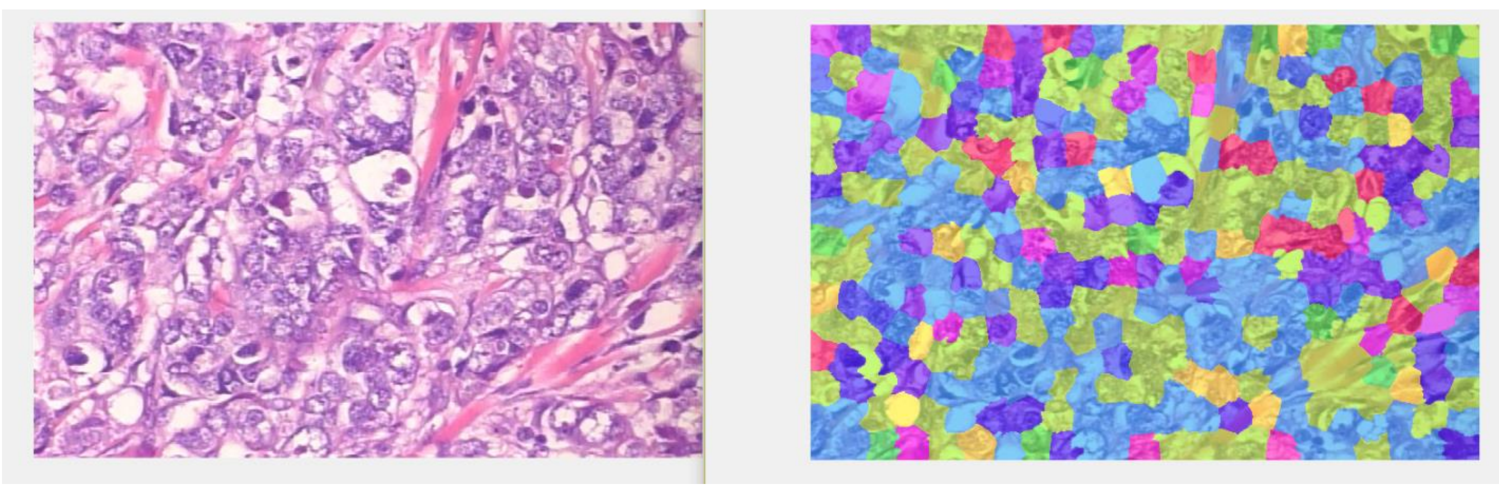

Figure 34. Image 4- Part 5 Result

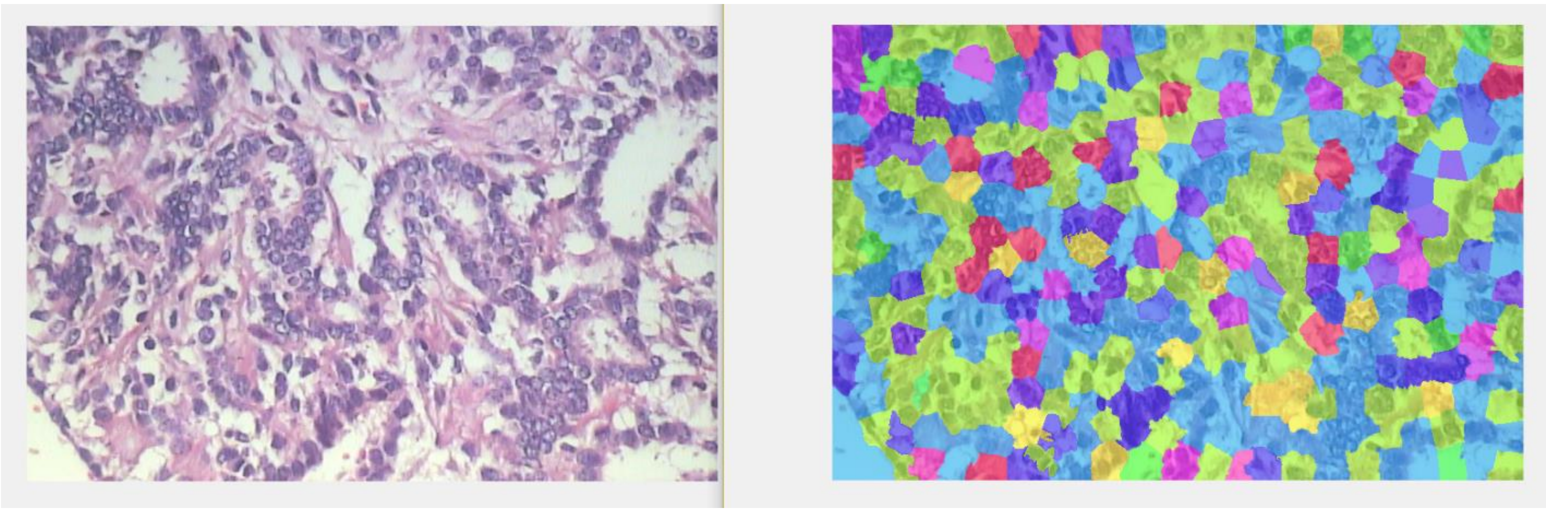

Figure 35. Image 5- Part 5 Result

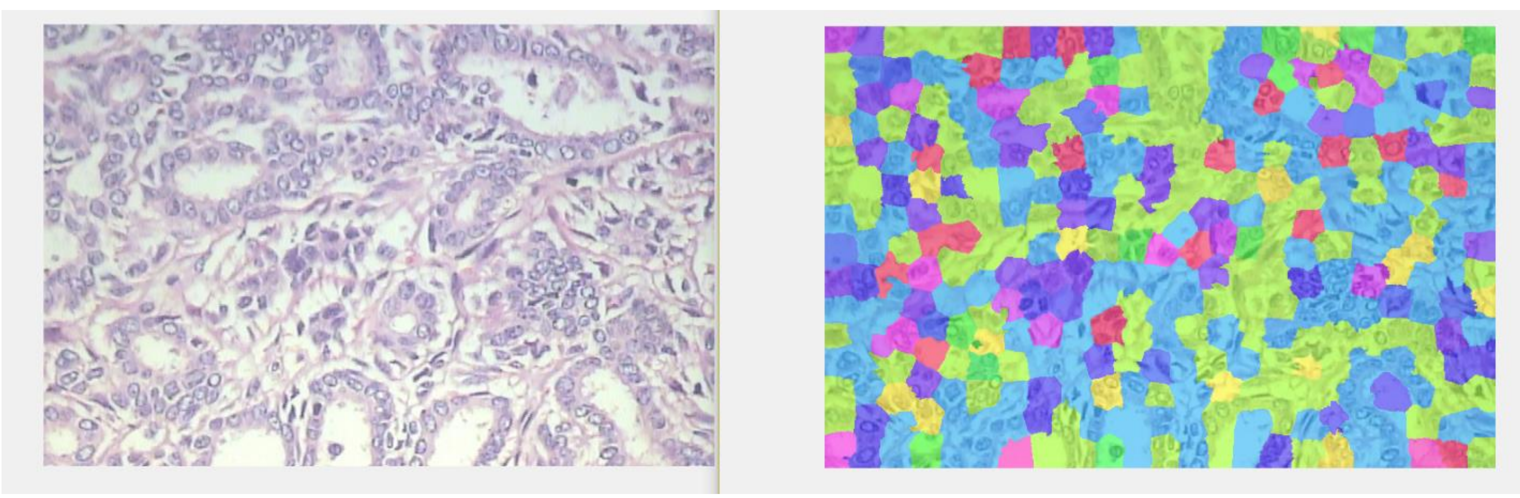

Figure 36. Image 6- Part 5 Result

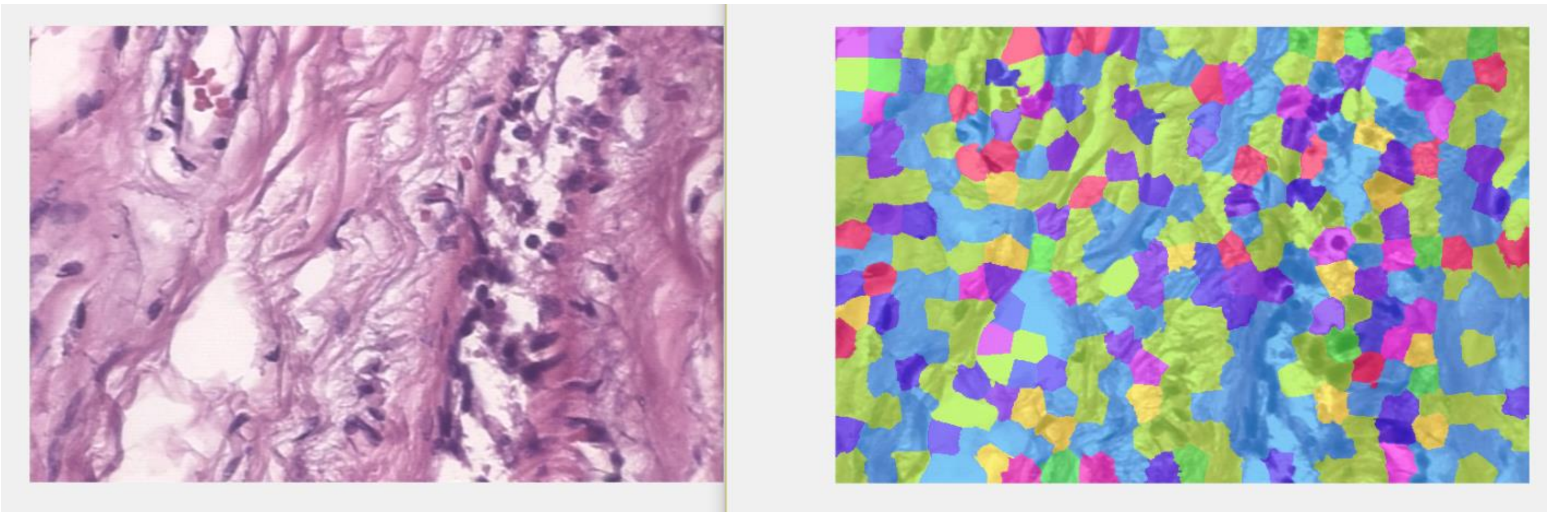

Figure 37. Image 7- Part 5 Result

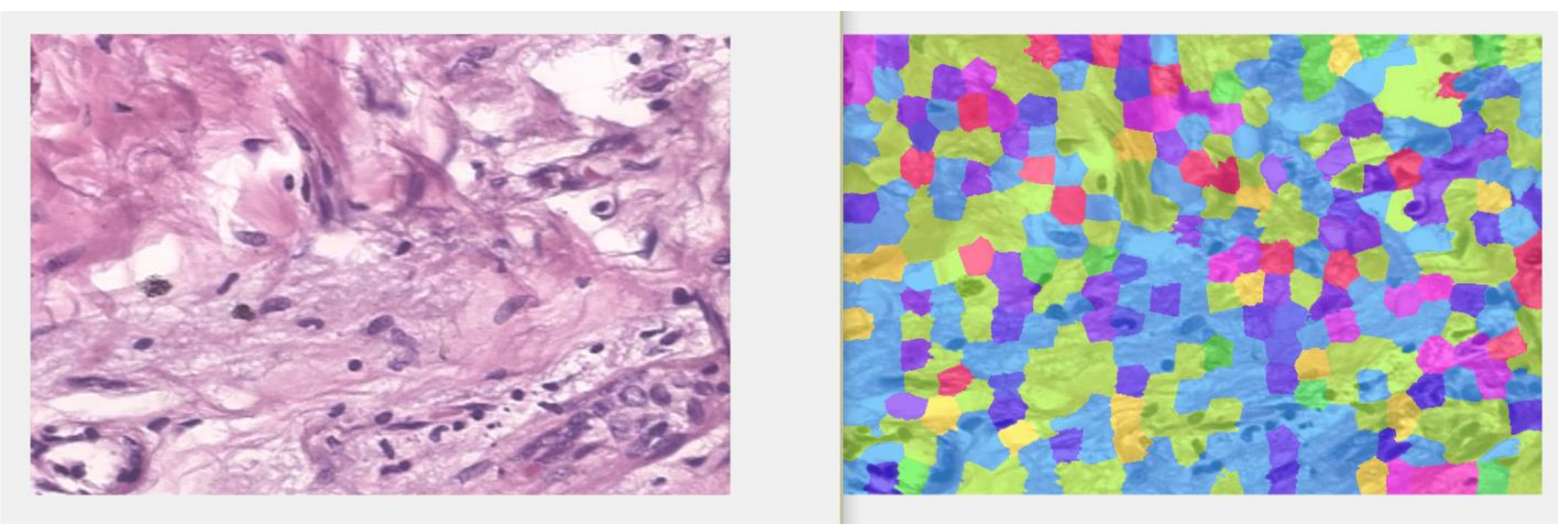

Figure 38. Image 8- Part 5 Result

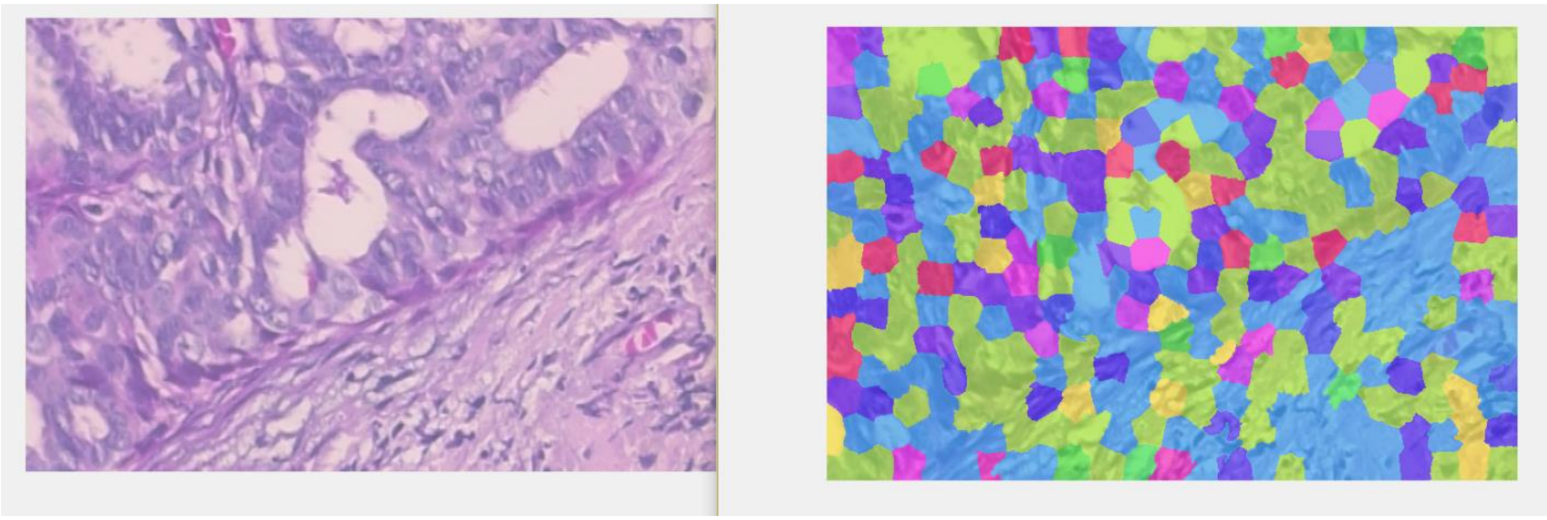

Figure 39. Image 9- Part 5 Result

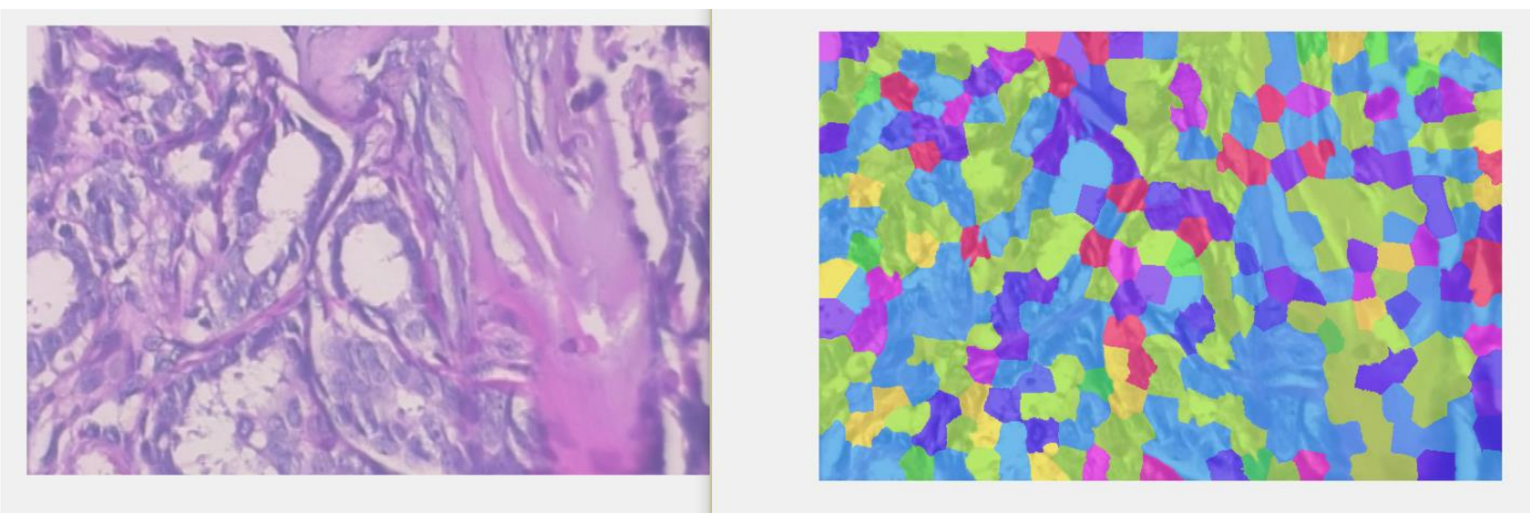

Figure 40. Image 10- Part 5 Result

8. Futher Analysis on Part 4 and Part 5
   Parameter Detection in Part 4:
   First Image vs. Last Image:

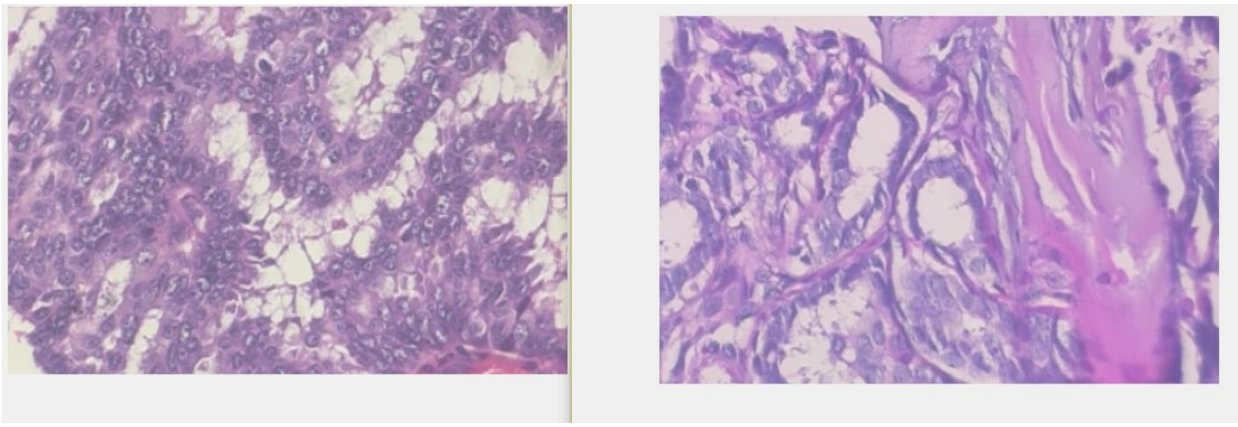

Figure 41. Image 1 and Image 2

The following parameters are used in the following results:
Superpixel size = 350, Gabor scales = [4 6 8 12], Number of clusters K = 10

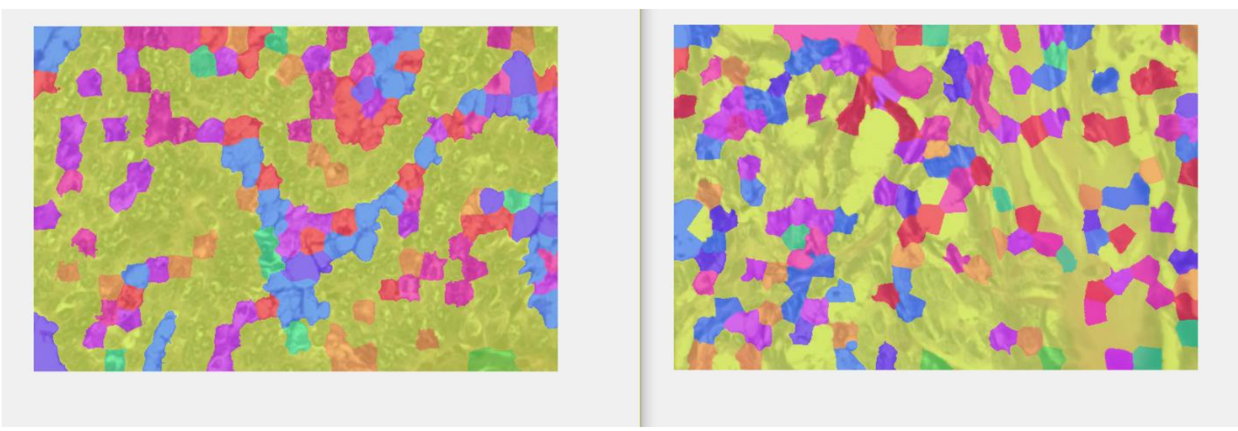

Figure 42. Image 1 and Image 2 False Color Representation

According to the above results obtained by using the above parameters, it can be seen that unrelated parts are clustered together. Hence, clustering parameter K seems to be too small. The yellow color in first image and the second image does not belong to same areas, in the first image, in the yellow area there are several nuclei, while in the second image, there are just a few.

Then only the K value is change and the number of clusters is incremented to 25.

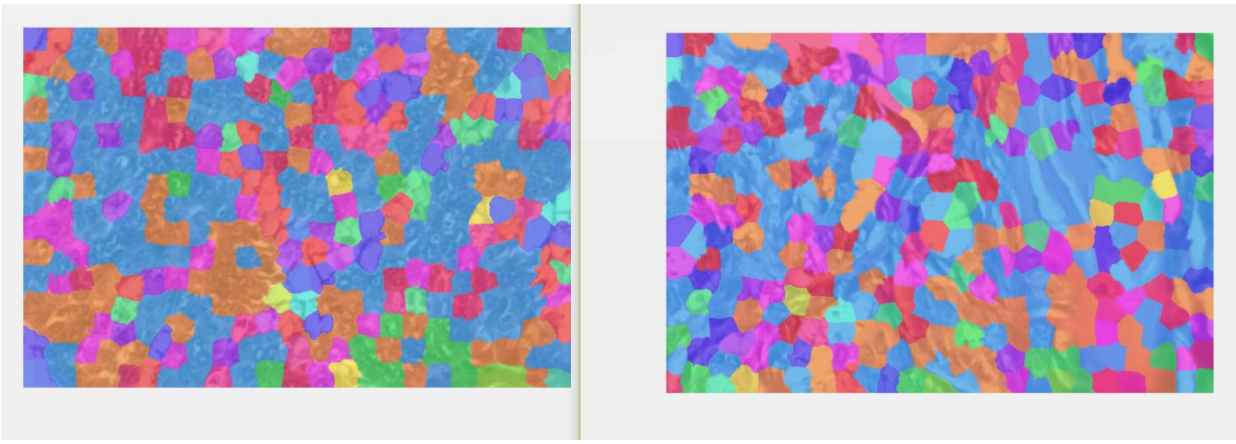

Figure 43. Image 1 and Image 2 False Color Representation

In these above results, some of the changes are not detected, therefore gabor scales are changed from [4 6 8 12] to [5 10 15 20]. However, in this case, the images were overclustered, therefore k value is selected as 20, and the results are as the following:

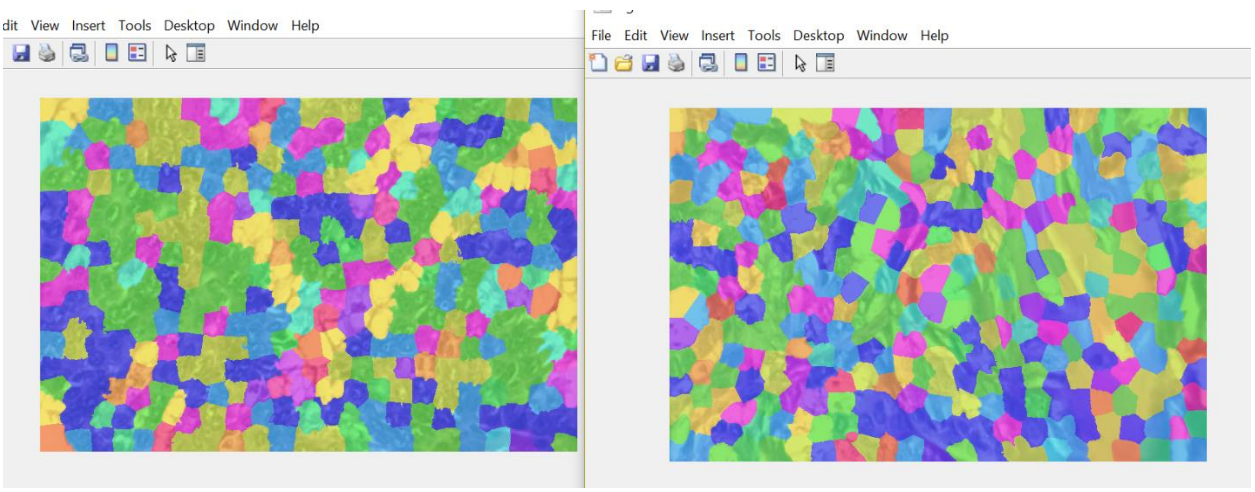

Figure 44. Image 1 and Image 2 False Color Representation

When number of pixels are dropped to 300, the clustering differs extremely as in the following figure.

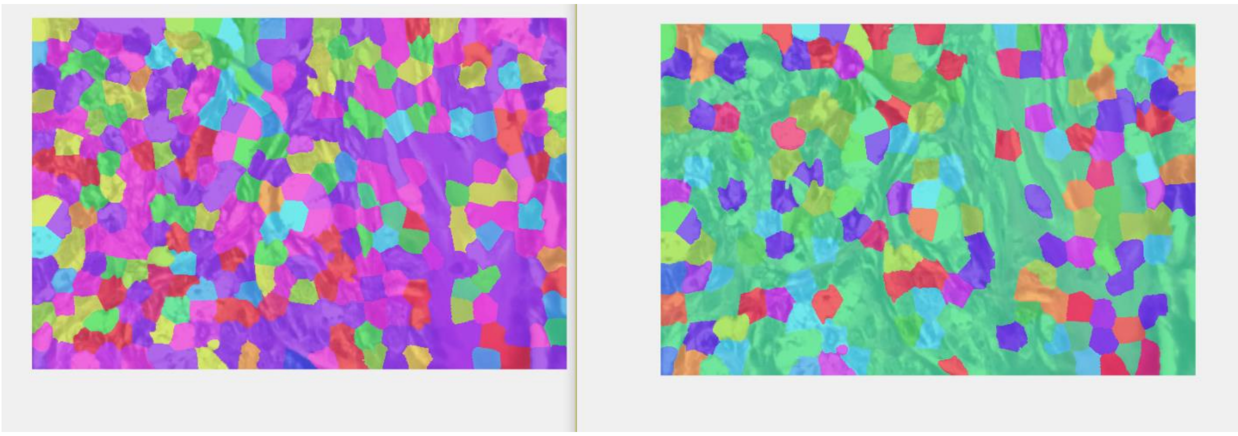

Figure 45. Image 1 and Image 2 False Color Representation

Different textures are not detected, nuclei and the tissues (white areas) are not differentiated at all.

When K value is incremented to 400 by using the same parameters, clustering seems to be inefficient. Therefore, the clustering parameter is incremented from 20 to 25, however, it does not provide better results.

In Part 5:

In this part the effect of radius selection is analyzed. R1 is used for describing the radius of first neighbourhood, while R2 is used for radius of second neighbourhood. Secondly, threshold , which is used while accepting a superpixel as a neighbour, is analyzed.

Firstly, R1 = 1.5 and R2 = 2.5 are selected.

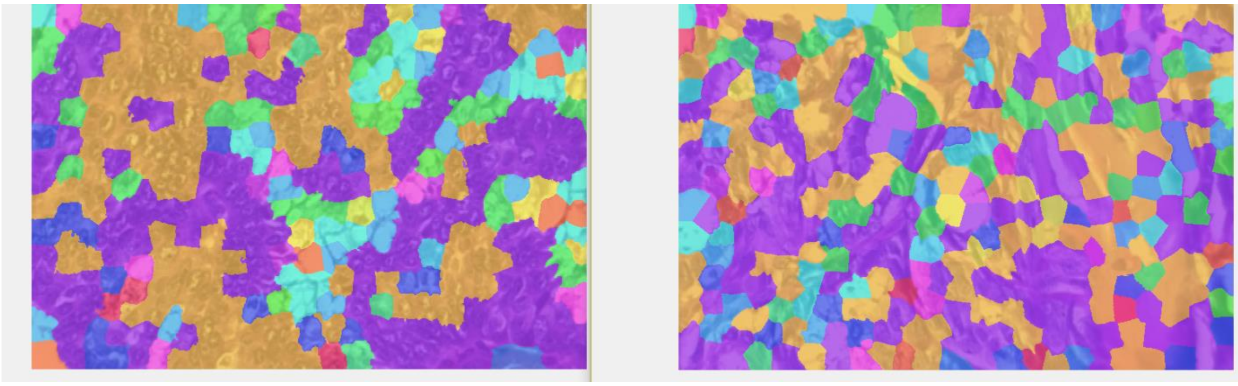

Figure 46. Image 1 and Image 2 False Color Representation

When R1 = 2, R2 = 3 the results are as the following:

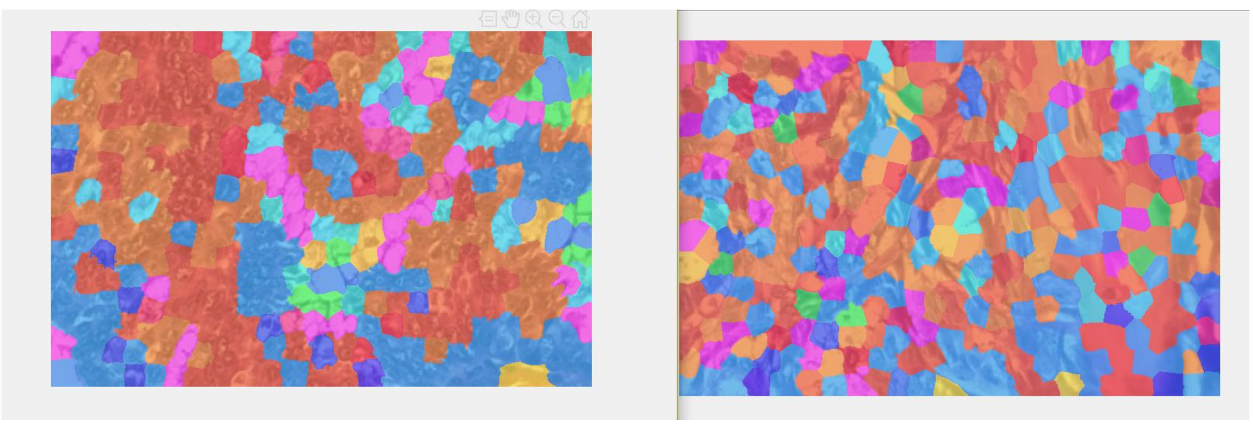

Figure 47. Image 1 and Image 2 False Color Representation

However, when R increments too much, clustering deteriorates. When R1 = 2.5 and R2 = 3.5, the results are the following:

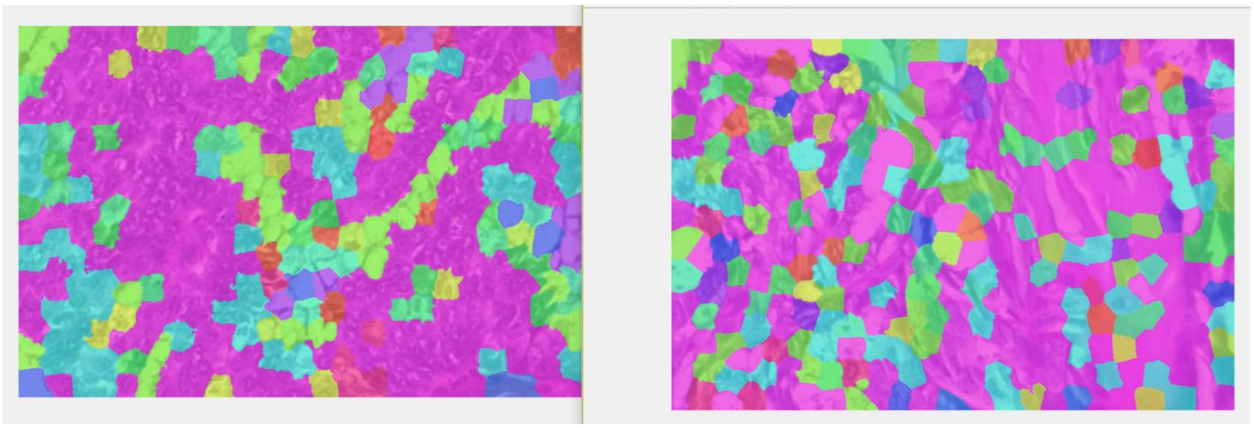

Figure 48. Image 1 and Image 2 False Color Representation

<u>Threshold value = 25:</u>

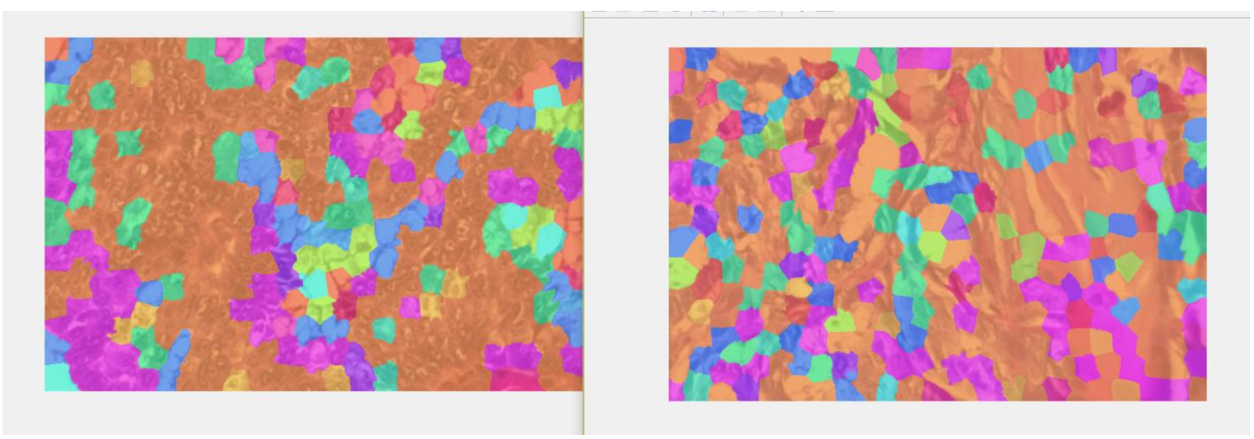

Figure 49. Image 1 and Image 2 False Color Representation

Threshold value = 50:

Figure 50. Image 1 and Image 2 False Color Representation

When threshold is incremented to 50, better clustering is achieved. Otherwise, when the threshold is set to 25, several neighbouring superpixels are accepted as neighbour and it deteriorates the clustering result.

9. Discussion:

In this assignment, feature extraction and image segmentation techniques are applied on the Breast Cancer Histopathological Database (BreakHis) in order to identify benign and malignant breast tumor tissues [1]. The database has H&E stained tissues and these above techniques helped us to detect nuclei and their coarse locations.

Gabor is used to identify textual differences. The images in the data set don't have patterns and they are not homogenous. They do not have uniform textures. The nuclei on those tissues are tried to be identified, therefore smaller wavelengths can bring more information about the nuclei. Several scales are selected and as a result of experimentation in order to detect small changes, texture more clearly, small scales are preferred. Up to 10(scaling factor) it detects small changes like the nuclei. After 10 as a scale factor, the results mostly show the white regions lines on the images. In order to differentiate both of them 5, 10, 15 and 20 are selected.

In part 4 of the assignment, the Gabor textures and color features are combined and according to these feature extractions of each superpixel, K-means clustering algorithm is used. As a result of this false color image representations are represented in Part 7.1. When the results are analyzed, it can be seen that, based on the shape and how they are stacked together some meaningful results are achieved. In Figure 21, the yellow colored regions do not have any nuclei. Furthermore, more clustered nuclei, which are close to each other, colored as green. The dark blue regions show the parts where nuclei overlap a lot and where they do not have a specific circular structure. The pink regions are used for tissues which surrounds the white circular shapes. According to this image, the feature extraction and clustering works well to identify different shaped and to identify different clustered nuclei. When Figure 22 is compared with the first image(Figure 21), it can be seen that there is some false clustering. For instance, the white regions are clustered in dark blue areas. Though it detects the nuclei which does not have circular shape, it accepts the white regions into this cluster as well. Additionally, it applies for green areas as well. In this picture, the small white and circular regions are clustered into the same group with more clustered nuclei.
In Figure 23, the nuclei are smaller and the white regions are more circular and smaller than other images. The dark blue region detects overlapping nuclei again at the right bottom corner. At the middle left area of the image, green regions detect small darker nuclei a lot, though it accepts some white regions. The shapes and the clustering of nuclei differs a lot in Figure 24, therefore more colors are represented. At the right bottom corner dark blue represents the same shaped and stacked nuclei. In Figure 25, the spacing between nuclei changes a lot, therefore it might give different Gabor features a lot therefore, more color represent in this picture. In Figure 27, the bottom dark blue regions select the weird shaped nuclei region in dark blue again. Besides, it selects the one alone nucleus at the right into a different region and clusters it into pink. In Figure 28, we can see the nuclei are clustered in dark blue region and again they do not have a uniform shape. In last two images, the pictures are more blurred, they probably have less resolution, it has some effects on clustering. Probably its Gabor texture representation is different than others. Furthermore, the tissue is different than others, it has more pinkish regions on the image, it does not cluster well those parts.

In Part 5, two level neighbouring algorithm is used, therefore now superpixels are represented with its own representation as well as its first and second neighbourhoods. Now, we have more features to represent a superpixel, hence it should provide a better clustering. In light blue areas, the number of nuclei is more than the other parts of the image and they are mostly more compact. In yellow regions, more circular and distinct nuclei present. They are dense and compact again. In Figure 35 at the middle bottom of the image and in Figure 37 at the left, dark blue region selects the weird shaped clustered nuclei. As explained before, two last images are more blurred, and there is not strict separations between nuclei. In the last image, the purple areas, have nuclei which seems to be continuous, it might be a higher level of cancer cell, the shape of it differs and it larger than the others. In this part, though it detects nuclei, it does not separate well the white regions from the regions from nuclei. This might be due to the fact that radius for first and second neighbourhood might not be selected perfectly. However, it seems to cluster similar nuclei in the same regions more accurately.

To sum up, the clustering results does not seem to be perfect in order to identify the benign and malignant tissues. It might be caused due to the complex nature of image scenes, as there are several touching and overlapping nuclei in the image. Additionally, the non-homogeneity of cells/nuclei causes more difficulty to find perfect fixed parameters for the whole data set. Furthermore, the H&E stained technique might not be good enough for us to separate cells from its surrounding. In order to improve the detection of benign and malignant tissues machine learning can be used, therefore by training better detection can be accomplished.

REFERENCE

[1] https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis/

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, pp. 2274–2282, May 2012.

[3] https://www.mathworks.com/help/images/ref/regionprops.html

[4] https://www.mathworks.com/matlabcentral/answers/169093-number-of-pixels-inside-circles-of-different-radius-on-a-single-picture#comment_261946