Fatemeh Darbehani [redacted]

ENSC474 – SFU – Spring 2017

Final Project

Overview

- In this project, we were given an Optical Coherence Tomography (OCT) scanning of the retina as shown below.

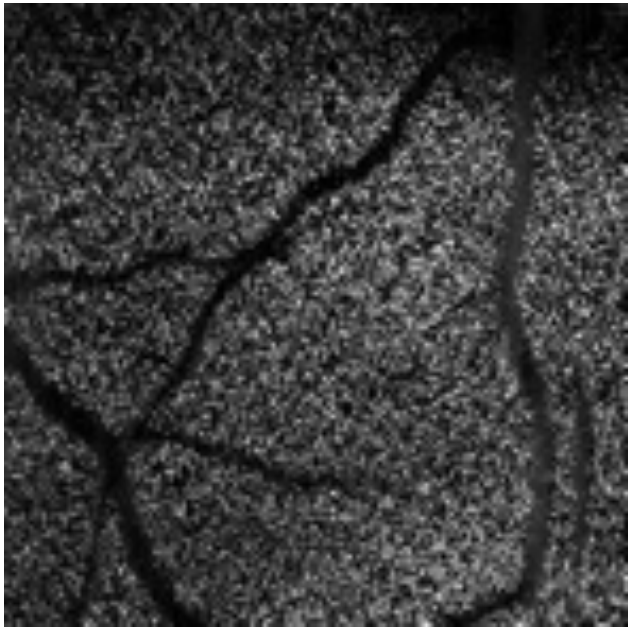


Figure1: Optical Coherence Tomography (OCT) of the Retina

- In the above figure each little white dot is a single photoreceptor in the retina.
- Our task was to count the number of photoreceptors and create a quantity map to present the quantity map in a local region.
- We were also asked to simulate a disease image in which some of these photoreceptors are randomly selected and destroyed in a localized region, mimicking the effect of a disease.

First Step

- This task is in nature a segmentation problem, grouping and distinguishing the white dots from the background. To segment based on intensity values my first step was to look at the histogram of the image and try to find the best threshold by observation.

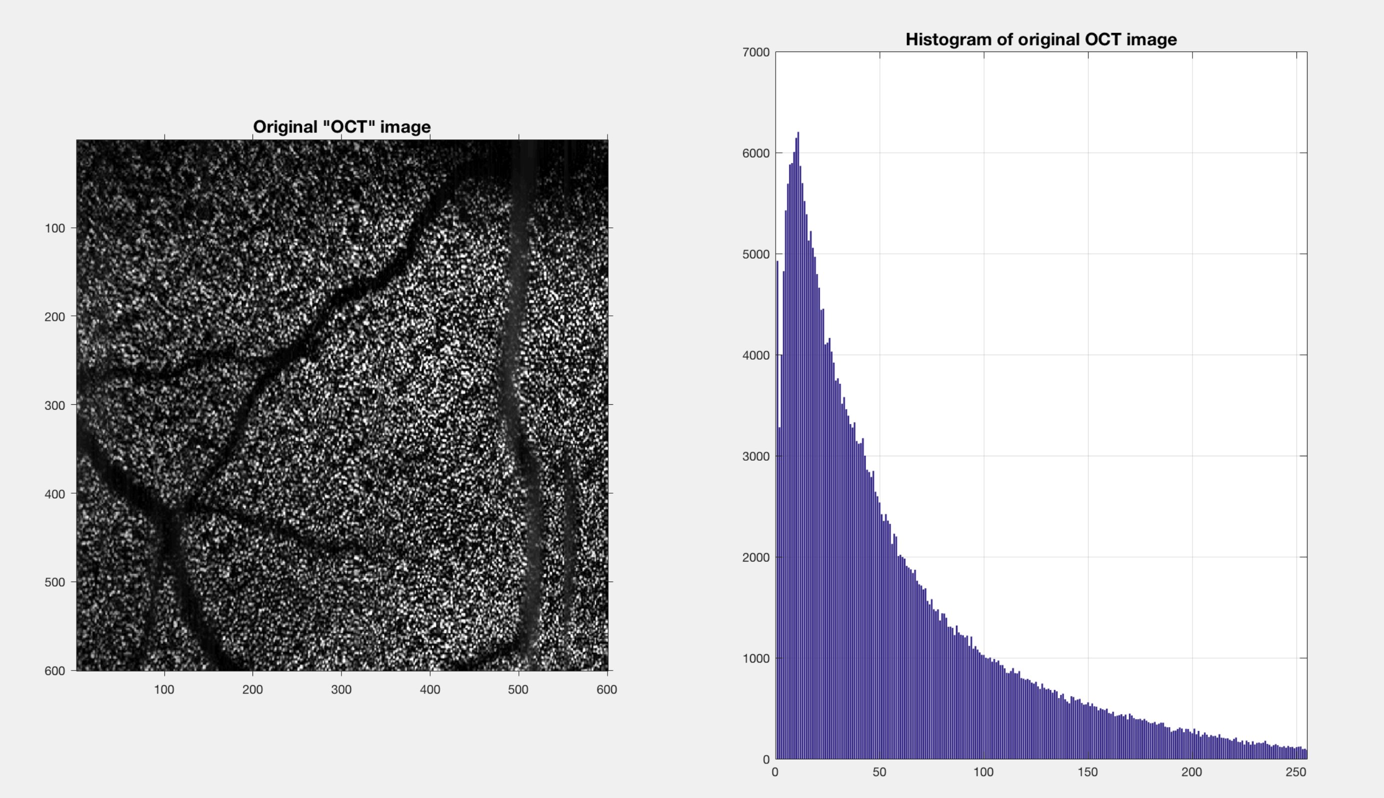


Figure2: Histogram of the Original OCT Image

Speckle Removal

- As can be seen in figure 2, there is only one peak in the histogram and there is no trivial threshold value that can be chosen.
- I thought this might be due to presence of noise in the image. After doing some research on the noise that exists in Optical Coherence Tomography, I found out that the noise that corrupts these images is called “Speckle” and the distribution of it can be represented with a Rayleigh distribution. This can be confirmed from the histogram of the image as well.
- Also, I found out that speckle is considered as multiplicative noise as opposed to additive noise which is the type of noises we mostly dealt with in the course.
- I tried to apply a logarithmic operator on the image to turn multiplicative noise into additive noise and perform low pass filtering to remove the speckles.
- Unfortunately, this did not result in improving the histogram of the image and I decided to process the image without trying to remove noise.

[Source: Baghaie A., Yu Z., D’Souza R. M., “State-of-the-art in retinal optical coherence tomography image analysis,” Quant. Imaging Med. Surg. 5, 603 (2015).

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4559975/]

Manual Threshold

- The simplest way of segmenting an image is global thresholding. I tried different values for the threshold and picked intensity of “100” to turn the gray-scale OCT image into a black and white image. The result is shown below.

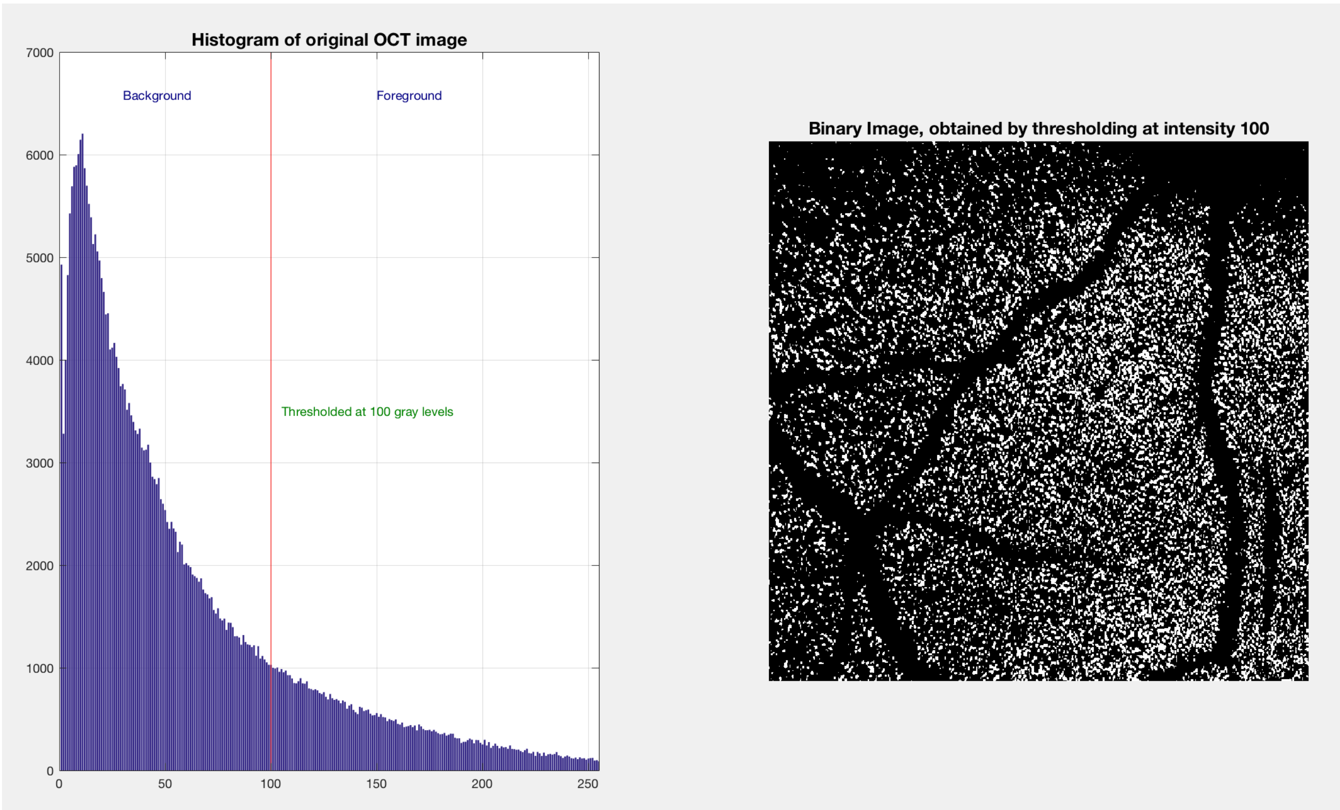


Figure3: Binary image thresholded at 100 grayscale level

ManualThreshold

- Although the above method is very easy to perform, the result will not be so accurate as the value of the threshold is image dependent and it is very likely to lose information in the darker or lighter areas of the image.
- Note that the binary image obtained from manual thresholding, could not segment photoreceptors in the darker or lighter areas of the image.

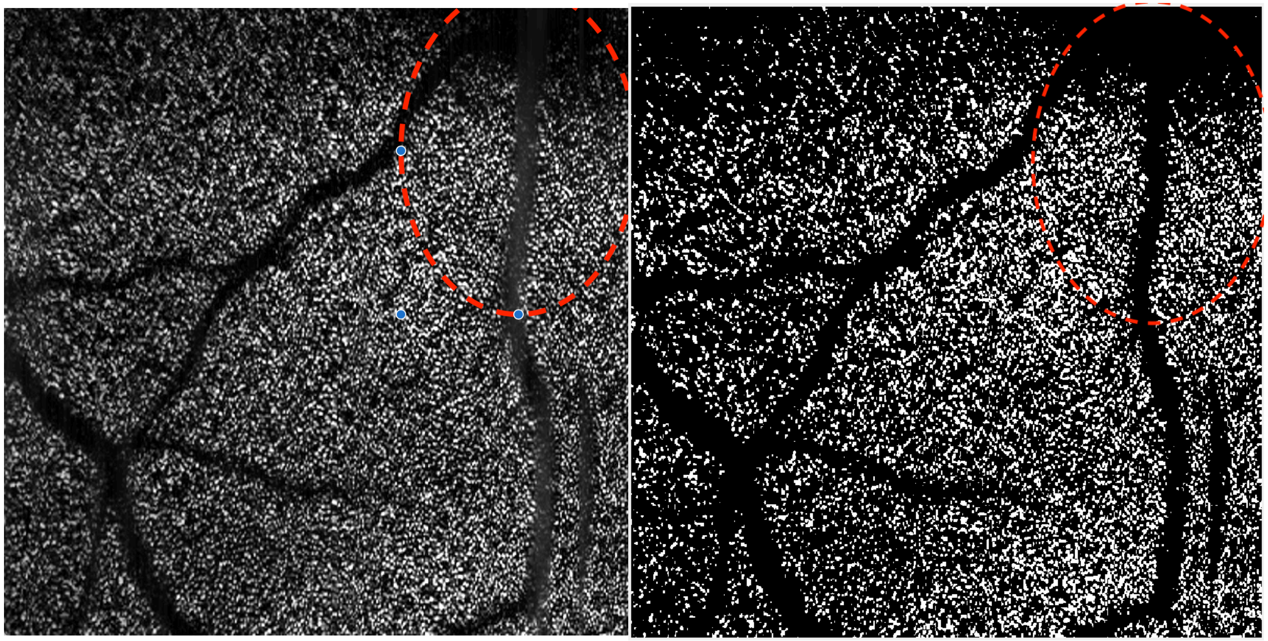


Figure4: Example of where global thresholding loses photoreceptors

K-means Clustering (2 clusters)

- A better way to choose a threshold as opposed to manually selecting a value is to use k-means clustering algorithm.
- K-means is a clustering method that aims to find the positions of the clusters that minimize the distance (Euclidean) from the data points to the cluster.
- The MATLAB inbuilt function “kmeans(X, k)”, performs k-means clustering to partition the observations of the n-by-p data matrix X into k clusters.
- I used this function to partition intensity values into 2 groups and created a binary image based on the which group each pixel belonged to. The result is shown below.

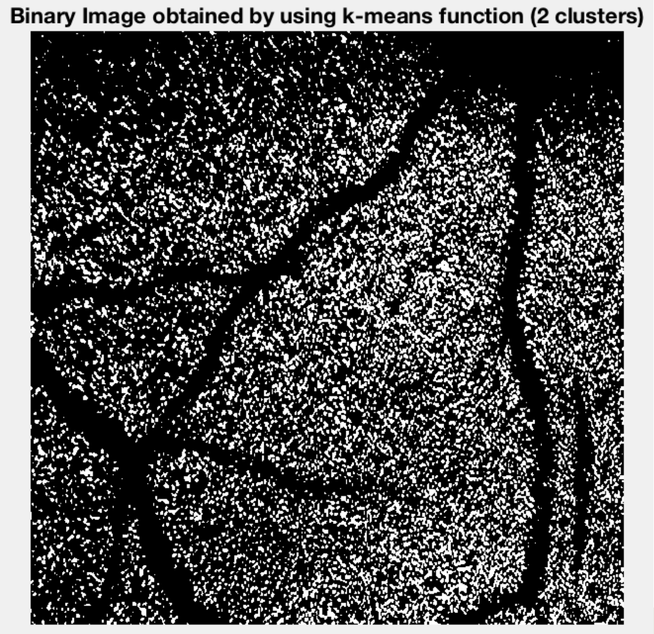
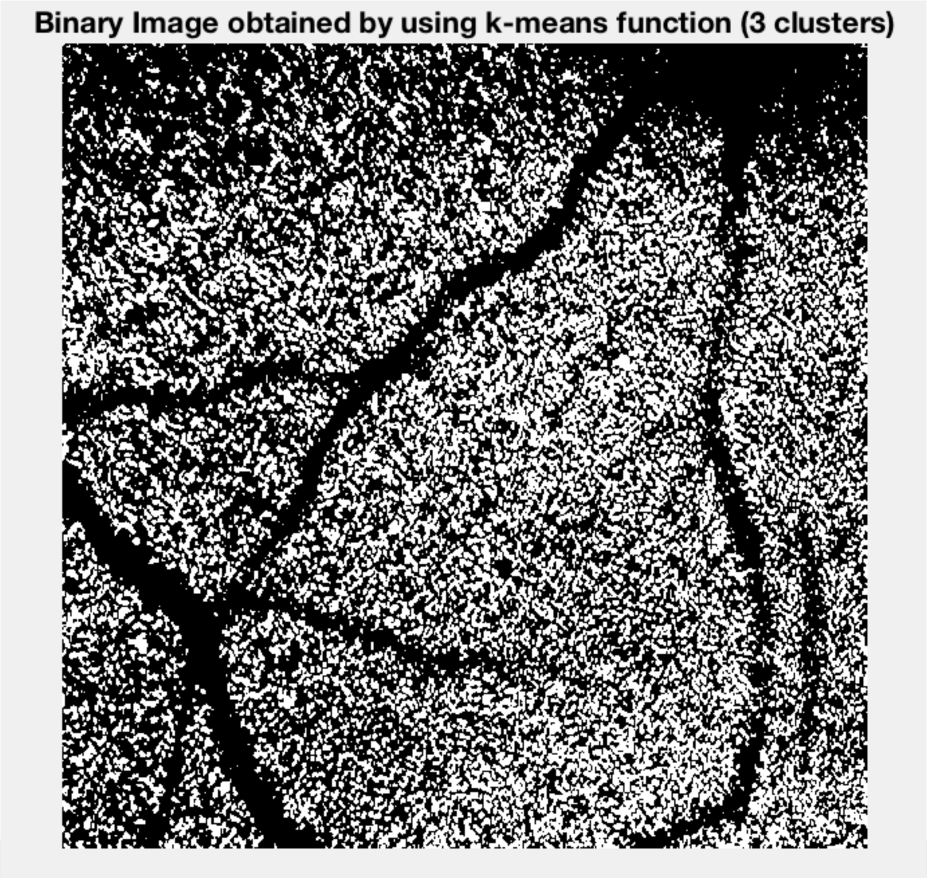


Figure5: Binary image using k-means (2 clusters)

K-means Clustering (3 clusters)

- As with the case of manual thresholding, k-means partitioning into 2 clusters was not able to detect photoreceptors that are not so bright or areas that foreground and background do not have much contrast.

- For that reason, I tried to cluster the pixel intensities into 3 groups to detect darker photoreceptors as well. Result of that is shown in the following figure.



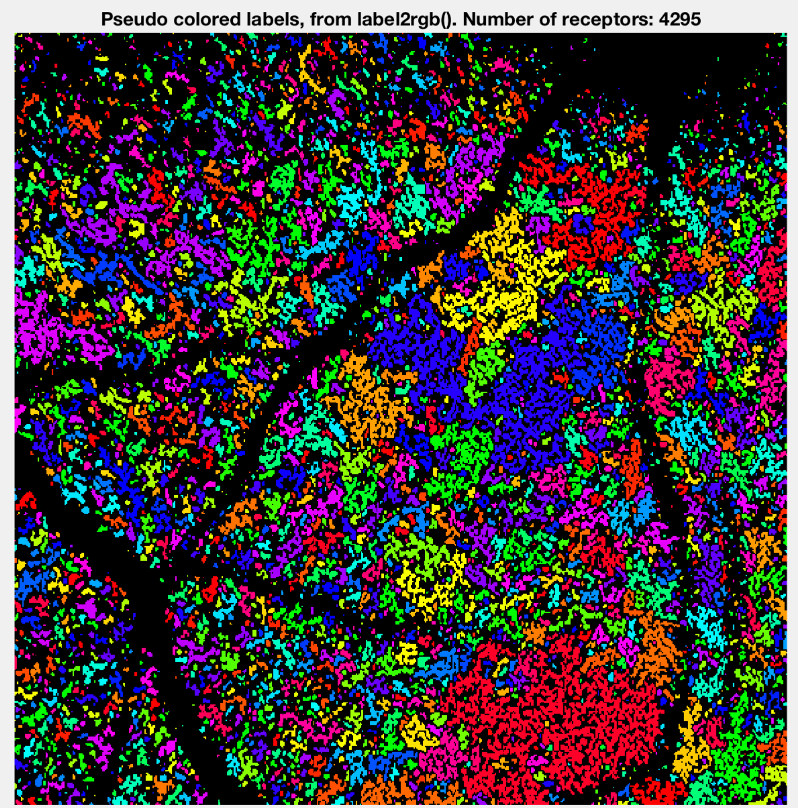
- From figure6, it can be seen that the binary image contains more photoreceptors compared to the two previous methods.

- However, the photoreceptors are grouped so closely to each other that it is hard to identify each photoreceptor correctly.

Figure6: 3 Clusters using k-means

Grouping the Photoreceptors

- I group the white dots by using the MATLAB built-in function “bwlabel”. It takes a black and white image and labels 4 connected objects in it.
- I chose 4-connected over 8-connected as the photoreceptors are in the forms of dots and they are more likely to be can only 4-connected than 8-connected.
- Then I assigned each object a random color and displayed the result.



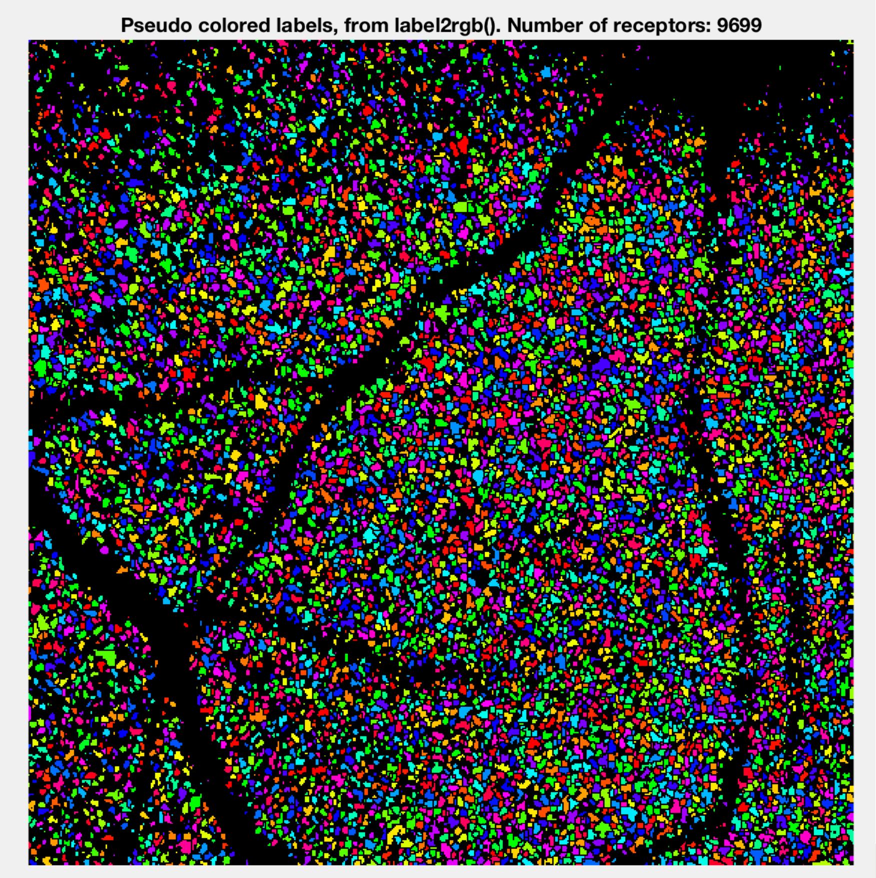
- Problem with this way of grouping is that a lot of photoreceptors are grouped together that results in undercounting their number.

Figure7: Grouped photoreceptors

Watershed segmentation

- I used watershed segmentation and distance transform to solve the problem of over merging the photoreceptors. The final image is shown below.

- In morphological image processing, the watershed transform is used to represent regions in a segmented image and the boundaries among them. In MATLAB, The inbuilt function “watershed” implements the watershed transform.



- The distance transform is a useful tool employed in conjunction with the water-shed transform. It computes the distance from every pixel to the nearest nonzero-valued pixel. It is implemented in MATLAB by function ”bwdist”, which allows specification of the distance method (Euclidean distance being the default) to be used.

- We can observe that the results is more acceptable and photoreceptors are grouped more appropriately

Figure8: Grouped photoreceptors using watershed segmentations and distance transform. Total number of photoreceptors: 9699.

Local Enhancement

- I noticed that the vein on the right of the image is lighter that the other veins and it made it harder for the white dots to be detected in that area.
- So, I created an enhancement function to enhance the intensity distribution of the that area. The function calculates the global mean and standard deviation of the image and compares that with local mean and standard deviation of each 5x5 block of the image.
- Because the area to enhance is lighter the the other areas, enhancement is done where the local mean is greater than 85% of the global mean and the local standard deviation is between 2% and 60% of the global standard deviation.
- If the area to be enhanced has a pixel intensity larger than 55, its intensity is multiplied by 1.5 and if the pixel intensity is less than 55, it is multiplied by 0.1.
- I picked all the above values by experiment. Below is the main part of this function.

```
%Build the local response.
if (Mloc>=k(1)*M) && (Dloc>=k(2)*D) && (Dloc<=k(3)*D)
    if Iloc(xc,yc) > 55
        g = 1.5 *Iloc(xc,yc);
    else
        g = 0.1 *Iloc(xc,yc);
    end
else
      g=Iloc(xc,yc);
end
```

Figure9: Part of the Local Enhancement Function

Result of Local Enhancement

- Result of applying the local enhancement function is shown in the following figure.

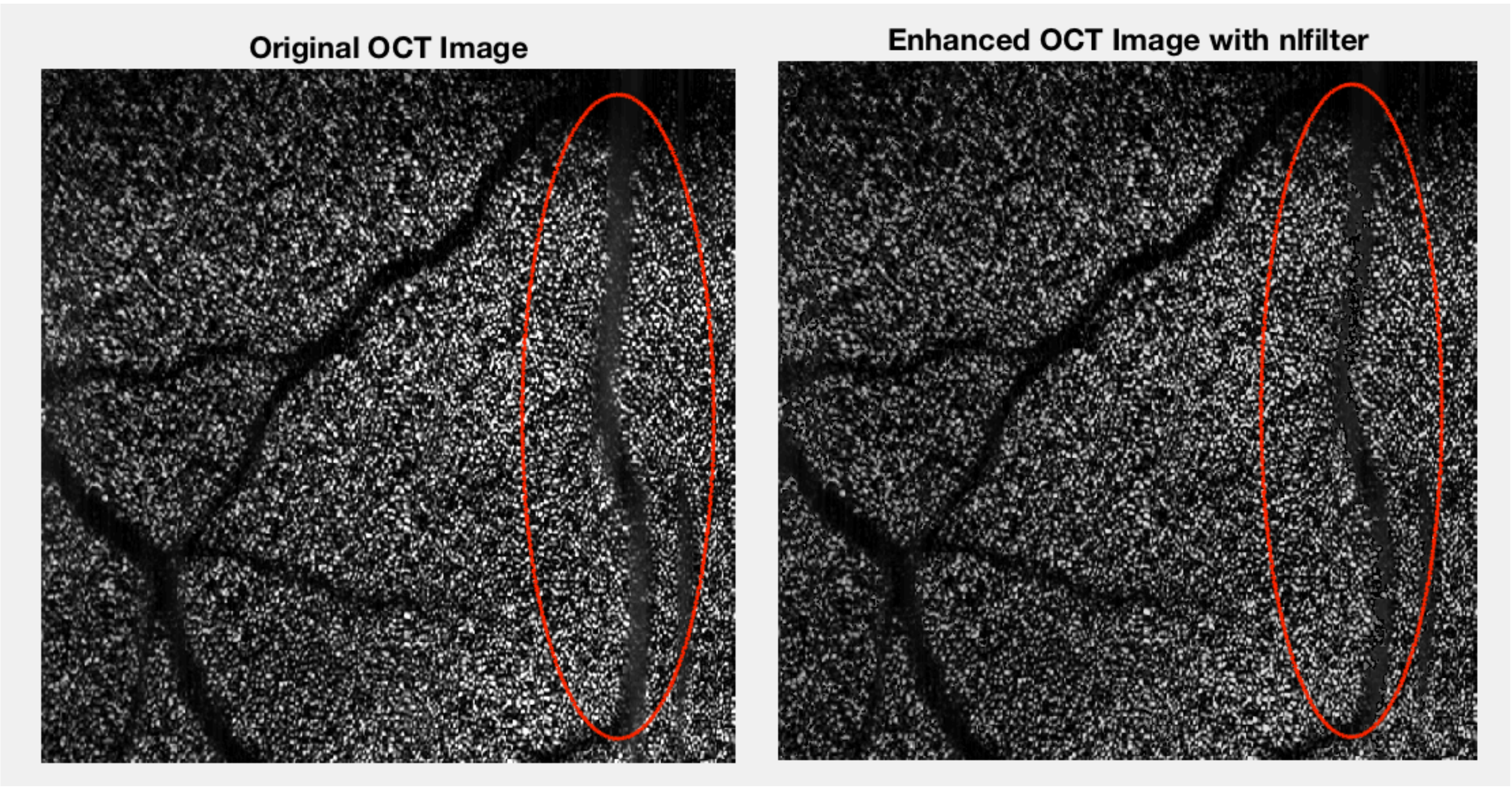


Figure10:Result of applying the local enhancement function

Result of Local Enhancement

- The histogram of the image then almost changed into 2 peaks.

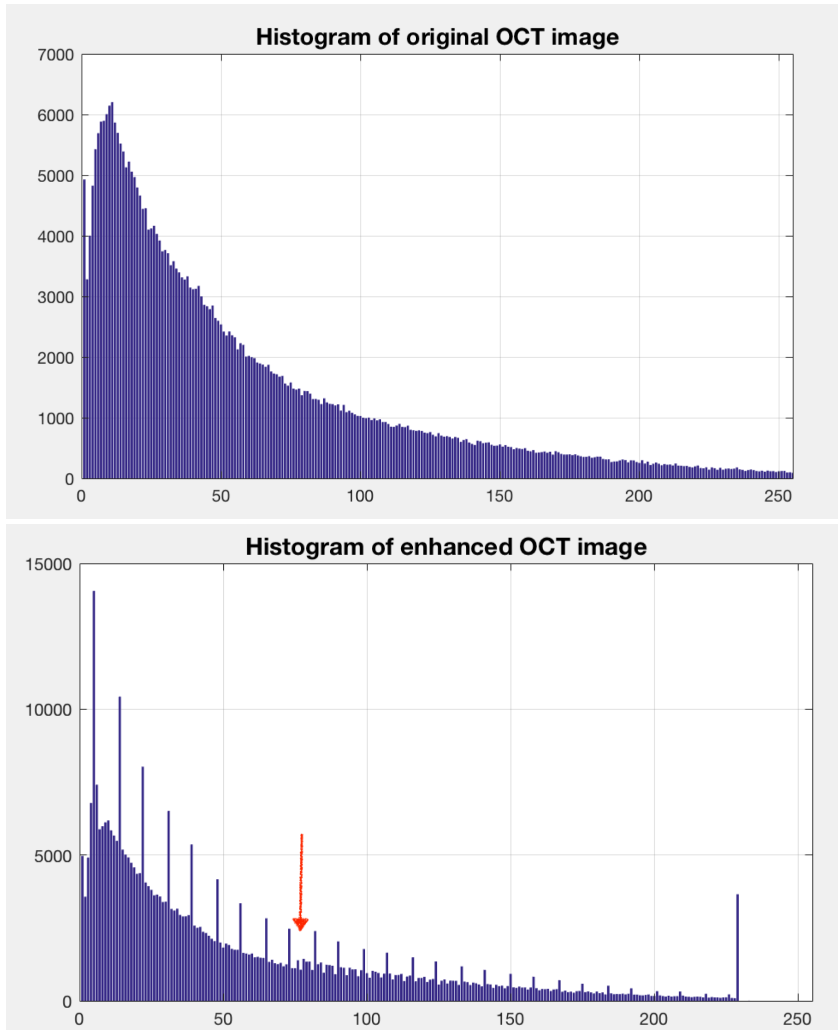
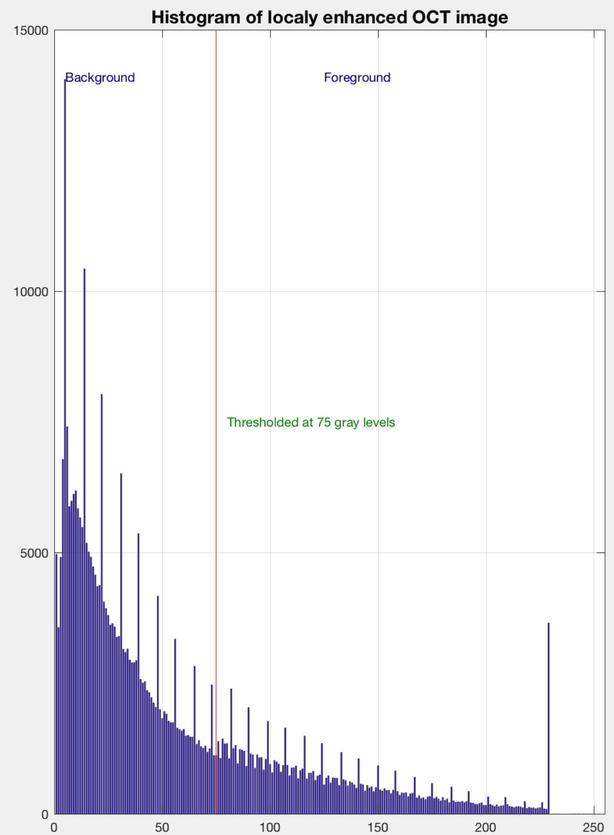


Figure11: Histogram of the Enhanced image

Manual Thresholding on the Enhanced Image

- I tried manual thresholding again with the threshold at intensity 75. Although the binary image has noticeably improved and detects more photoreceptors with comparison to the the first global thresholing, this method is still not desirable as it is image dependent.



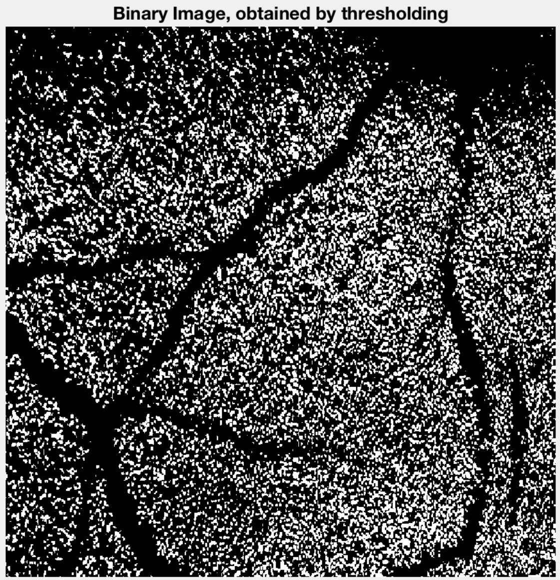
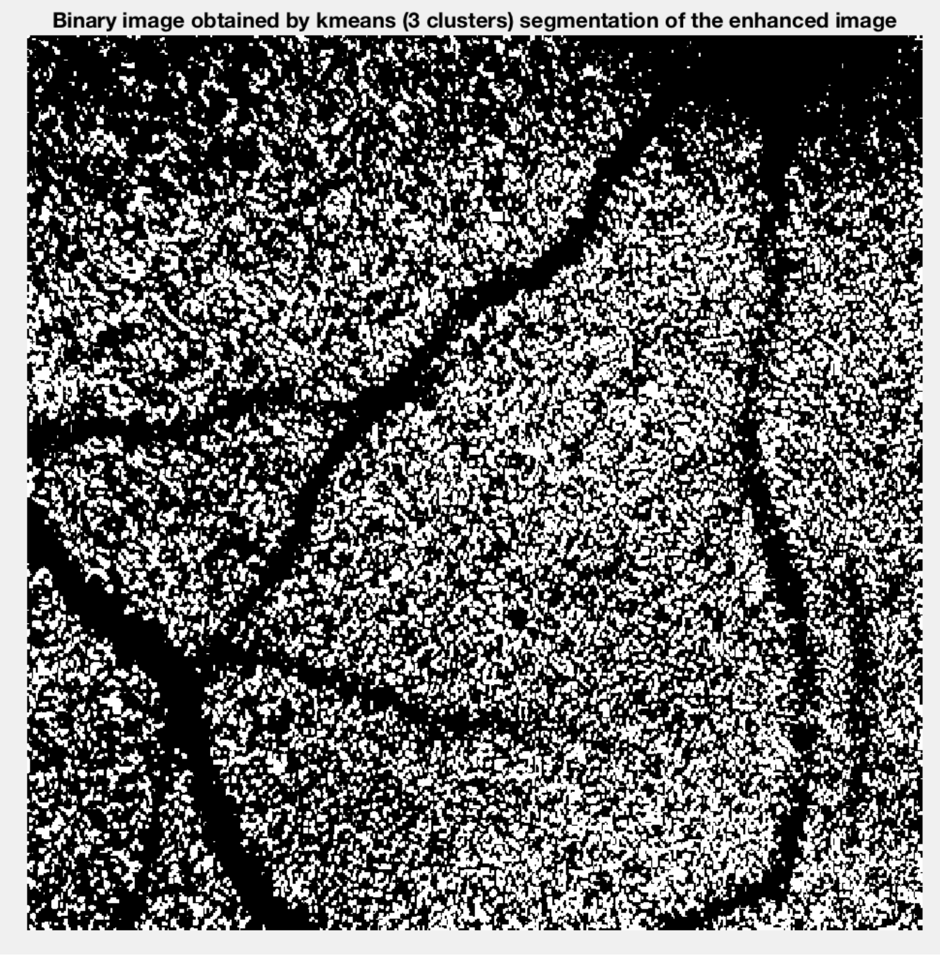


Figure12: Enhanced image thresholded at 75 grayscale level

K-means clustering on the Enhanced Image

- The following shows the result of 3 cluster k-means segmentation on the enhanced image.



- the results of both segmentation methods indicate that the local enhancement improved the result of segmenting the image and less photoreceptors were left undetected.

Figure13: 3 cluster k-means segmentation of the enhanced image.

K-means clustering on the Enhanced Image

- The following images show the result grouping and using watershed segmentation on the previous binary image.

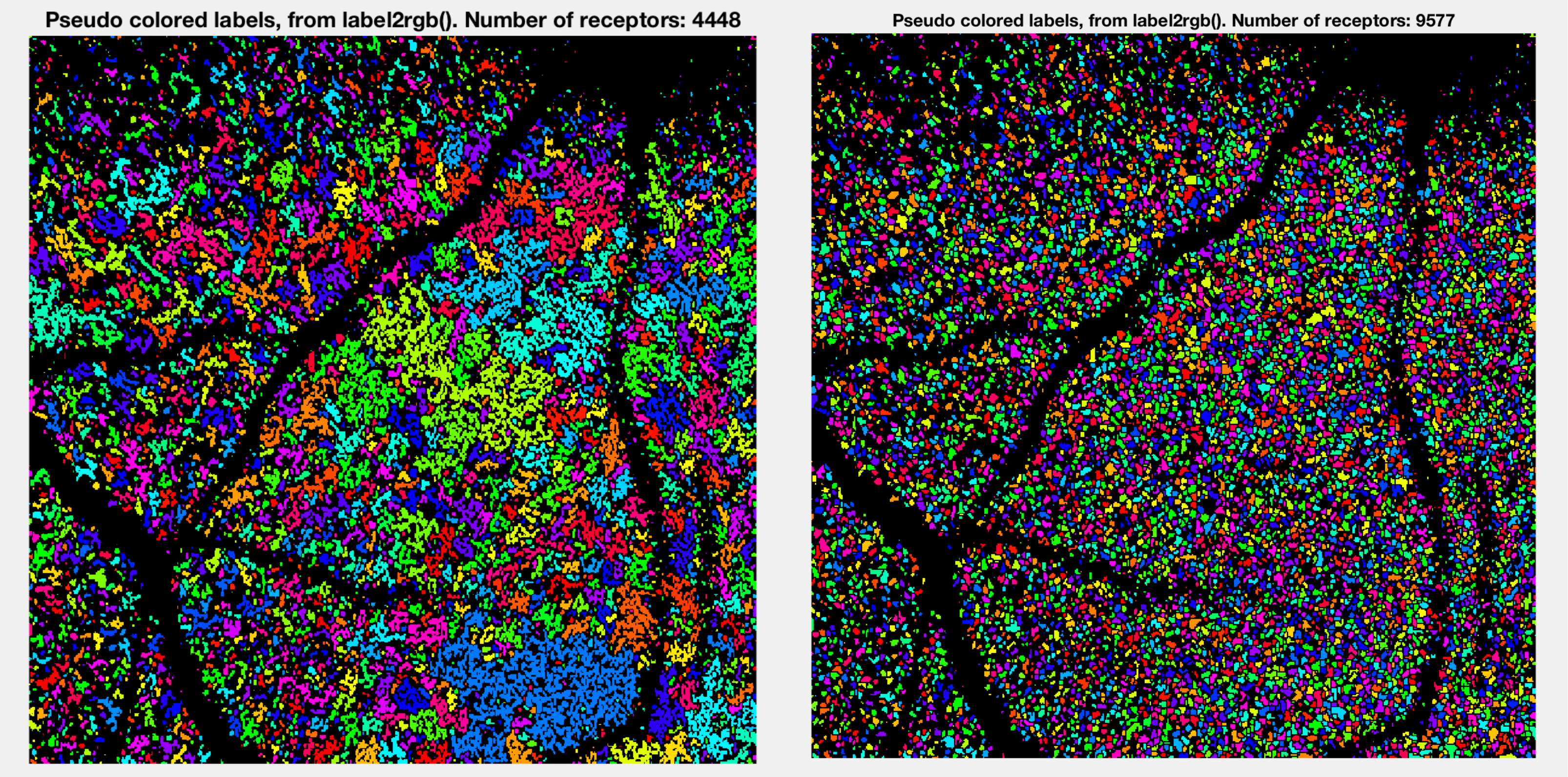


Figure14: Grouping photoreceptors. Total number of photoreceptors found: 9577.

Local Thresholding

- I also tried the MATLAB inbuilt functions "adoptthresh" and "imbinarize" with different values for sensitivity and size to locally threshold the image. But they all resulted in overcounting the photoreceptors. Result of local thresholding with sensitivity 0.3 and size 3x3 is shown below. As I increased the sensitivity and size, the overcounting increased.

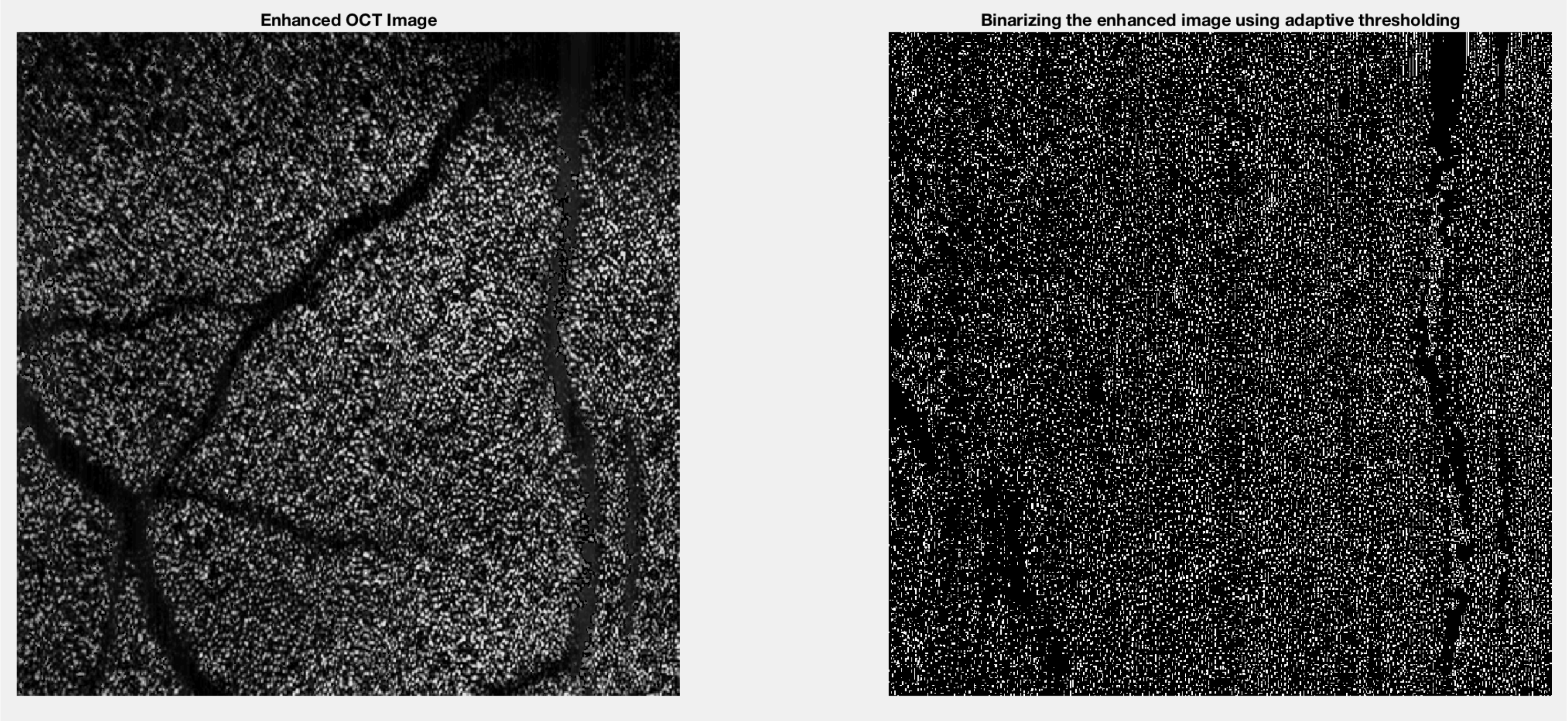


Figure15: Result of binarizing using local thresholding

Otsu's Method

- Finally I tried Otus's method by using the inbuilt functions "multithresh" and "imquantize" to segment the image.

- Otus's thresholding chooses the threshold to minimize between class variance of the the thresholded black and white pixels.

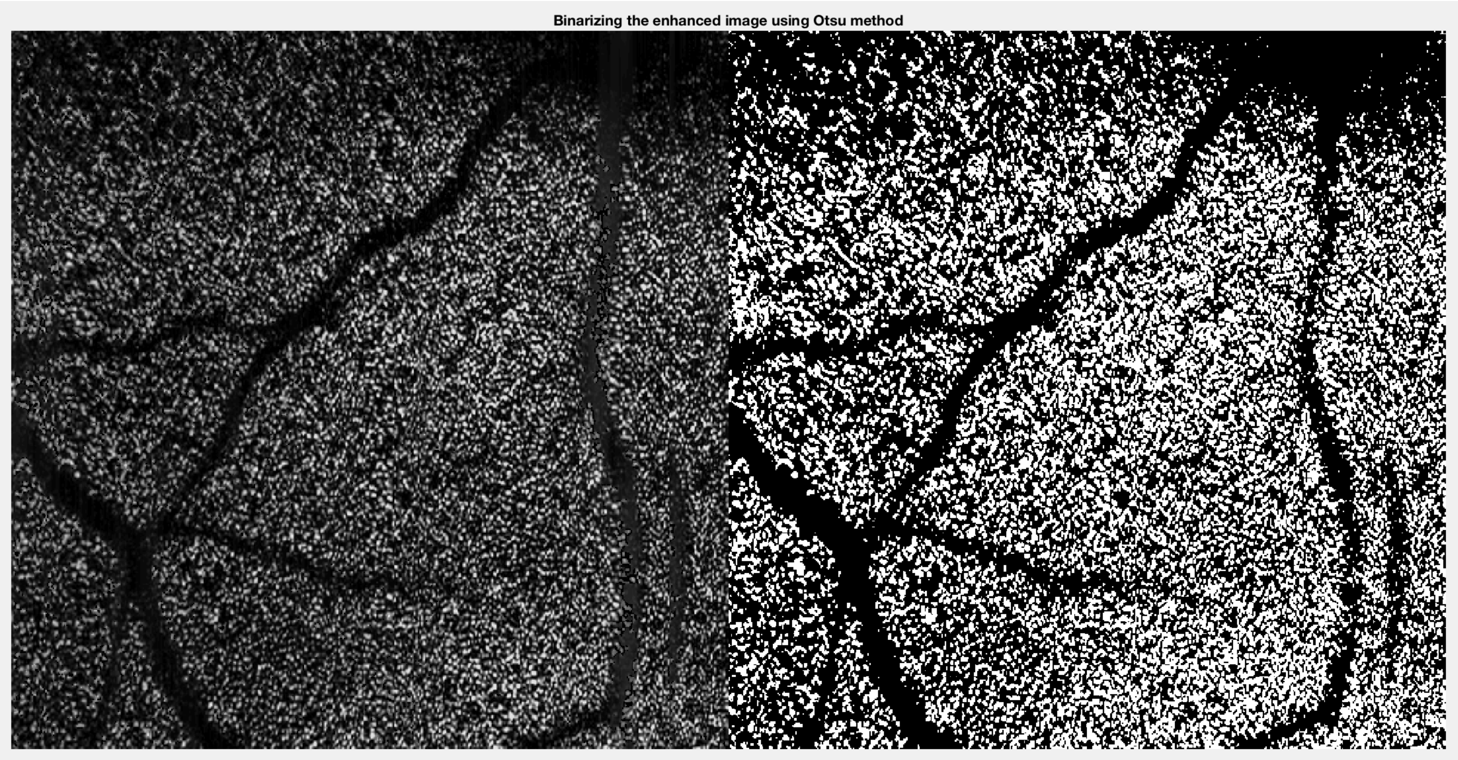


Figure16: Result of binarizing using Otsu's method

Otsu’s Method

- The following image shows the result grouping photoreceptors in the binary image obtained by using Otsu’s method.

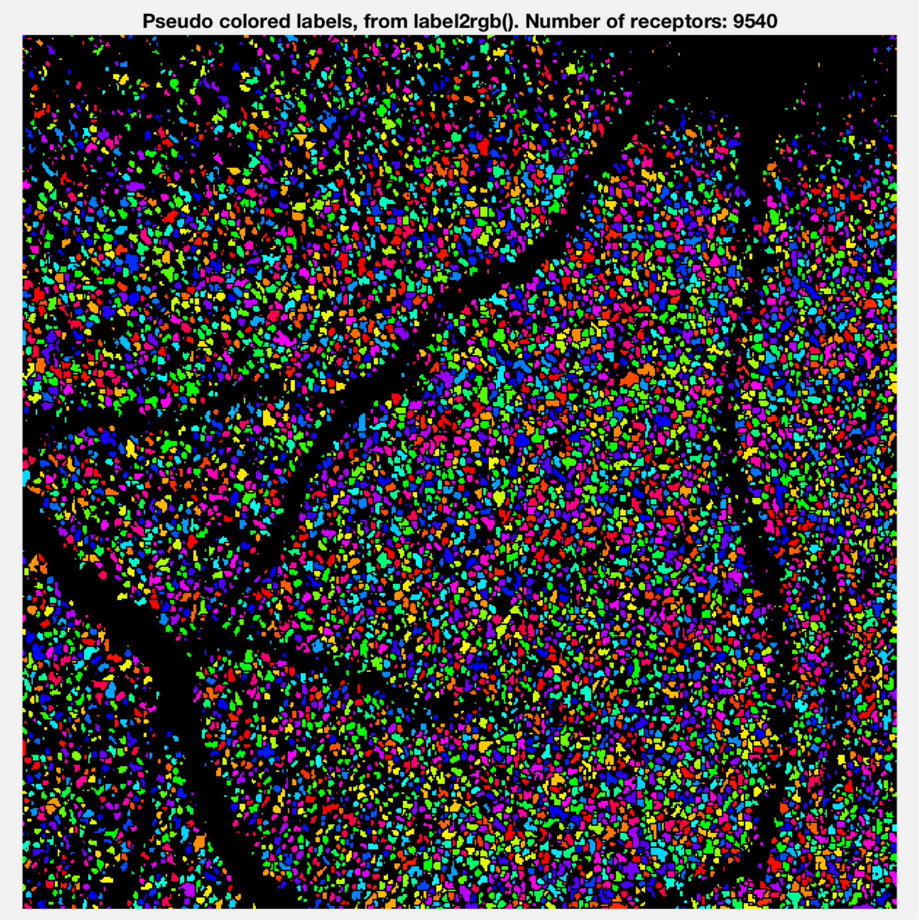


Figure17: Result of binarizing using Otsu’s method. Total number of photoreceptors found: 9540.

Global Thresholding Methods Limitations

- From the experimental results, the performance of the global thresholding techniques including Otsu’s method is shown to be limited by the small object size, the small mean difference, the large variances of the object and the background intensities, and the large amount of noise.

- In this project, due to small size of the foreground objects and some amount of noise present in the image, the histogram of the image doesn’t show bimodality and which results in segmentation error.

- Using Otsu’s method in this project, the means of the background and foreground were 51 and 125 respectively and because the difference between them was not too small, this method showed reliable result.

Preferred Method

- After trying multiple methods of thresholding, I observed that after locally enhancing the image, 3 cluster k-means and Otsu's methods both resulted in more accurate segmentation of the photoreceptors.
- I finally decided to use Otsu's method as it performs better in identifying photoreceptors in the darker areas of the image (upper corners of the image).
- Then, I created a program to prompt the user for a mask size with the default mask size of 25.
- After that, the program displays the OCT image and enables the user to select an area and shows the number of photoreceptors in the area of interest.

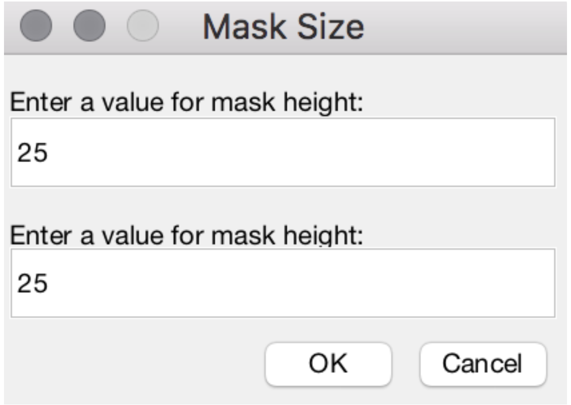


Figure18: Prompt window to set size of the quantity map

Program’s Interface

- Program’s interface is shown below. It lets user continuously select a region and print the number of photoreceptors in that region. It also colors the photoreceptors that are detected in that region.

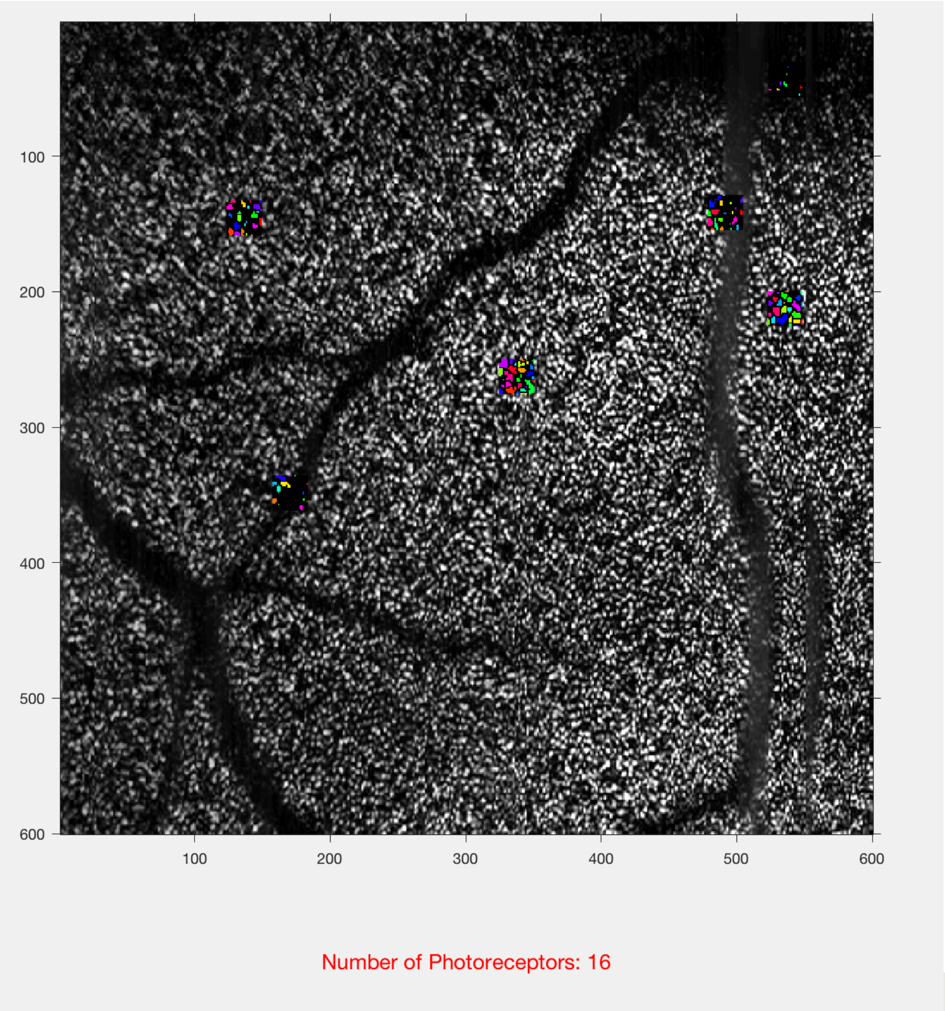
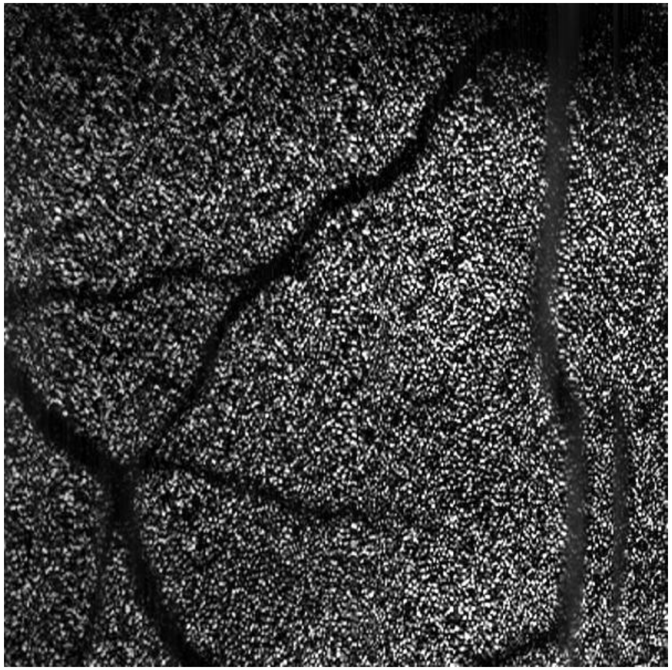


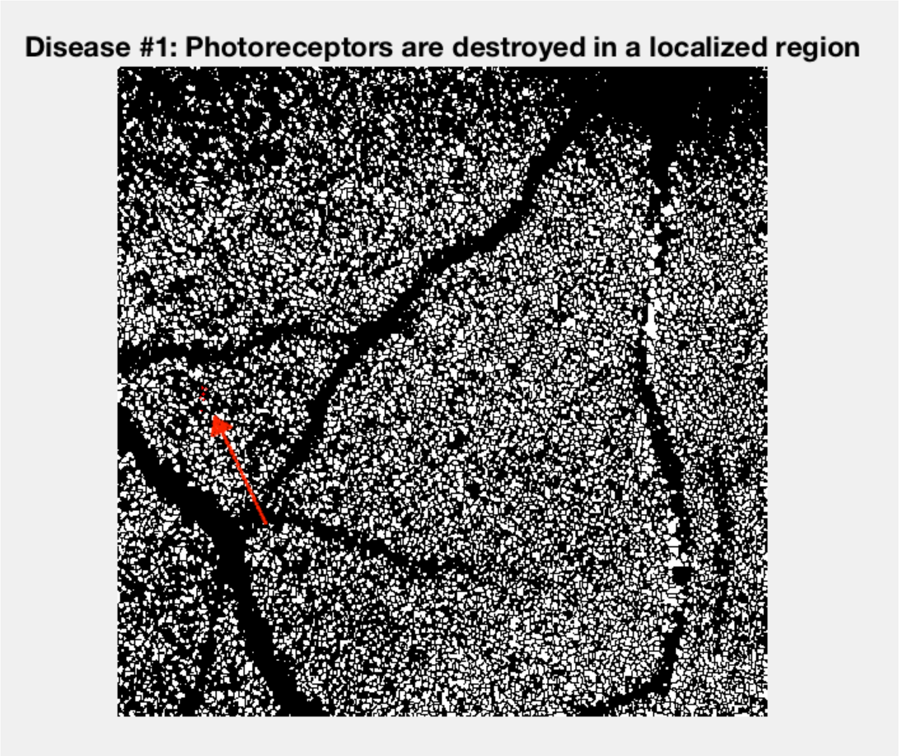
Figure19: Program’s main interface

Simulated diseases

- I simulated 4 images with disease and tested my program to count the number of lost photoreceptors. The lost photoreceptors are they shown by red dots on the image.
  - The first disease has destroyed the photoreceptors in a localized area.
  - The second disease has destroyed the photoreceptors in two areas.
  - And finally, the third disease has destroyed the photoreceptors in multiple areas.

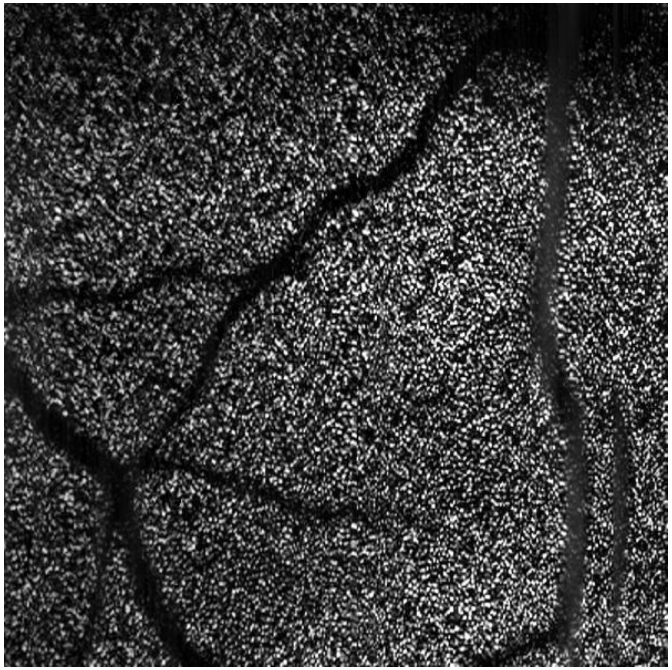
Figure20: Disease #1, 3 destroyed photoreceptors

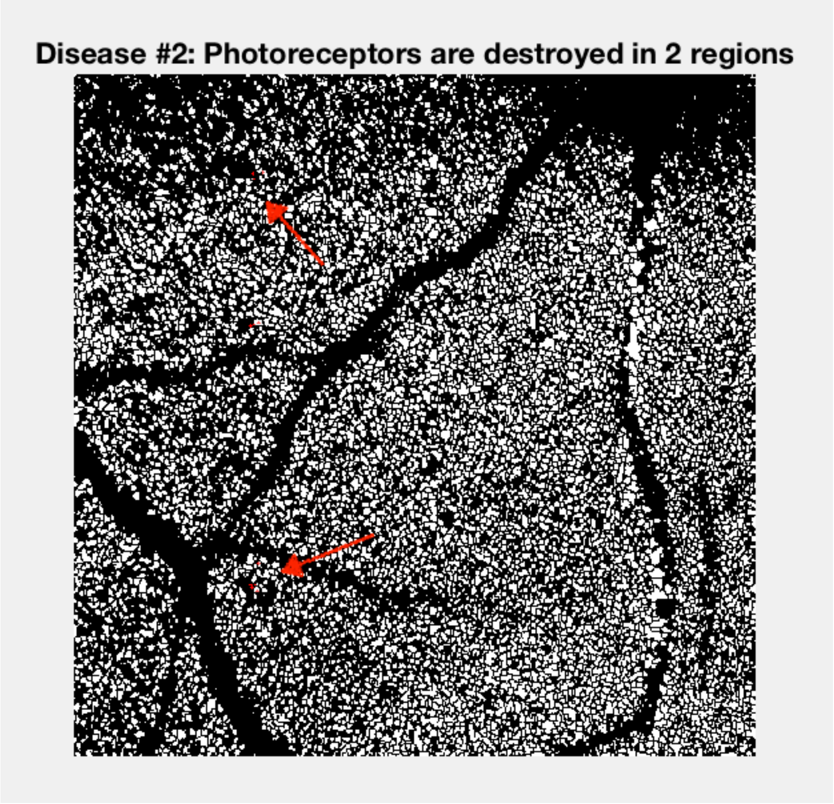




Simulated diseases

Figure21: Disease #2, 5 destroyed photoreceptors





Simulated diseases

Figure22: Disease #3, 20 destroyed photoreceptors