INTRODUCTION TO IMAGE PROCESSING

ASLI KURT

151220182068

The 8 steps in the assignment were completed by dividing them into 3 parts. These parts are as follows: Part 1 (1st and 2nd step), Part 2 (3rd, 4th and 5th step) and Part 3 (6th, 7th and 8th step)

Collecting the steps in parts was preferred to make it easier and clearer to solve the problem and to facilitate the evaluation of the results separately.

PART 1

First, the image file with .jpg extension named NEI-medialibrary-4506615 was read. This file is referred to as the "original image" throughout the work. The image read is an RGB image with a size of 564x564x3 pixels. The read image is displayed in figure form.

Then, the original image was converted to grayscale format using the rgb2gray function. The resulting grayscale image is displayed.

Then, the parameters required to implement the Canny Edge Detection operation were defined. These parameters consist of minimum threshold (Tlow), maximum threshold (Thigh) and sigma (sigma). After this process, Canny Edge Detector operation was applied. For this operation, the edge function in the MATLAB program was used. The result obtained from this study is displayed in figure form.

After these steps, optimum values were calculated for the values of the parameters used. Meanwhile, the value '2' was selected for the optimum sigma value (sigma_opt). Additionally, the graythresh function in the MATLAB program was used to find the optimum threshold values. The result obtained from this study is displayed in the form of a figure.

All images obtained throughout the study are included in the Part 1 in the form of figures, and the axis information of all images is included. Code blocks are available at the end of the part.

Implications

When the original image was converted to gray level, the blue fluorescently stained cell nuclei became much less visible compared to the green fluorescently stained cell walls. Therefore, cell walls are more prominent than cell nuclei. This significantly reduced the detection of cell nuclei. In such a case, applying the Canny Edge Detection operation detects only the cell walls. Even

if Sigma and Threshold values were changed, cell nuclei could not be detected. This can be prevented by gamma-functioning the image or adjusting the contrast.

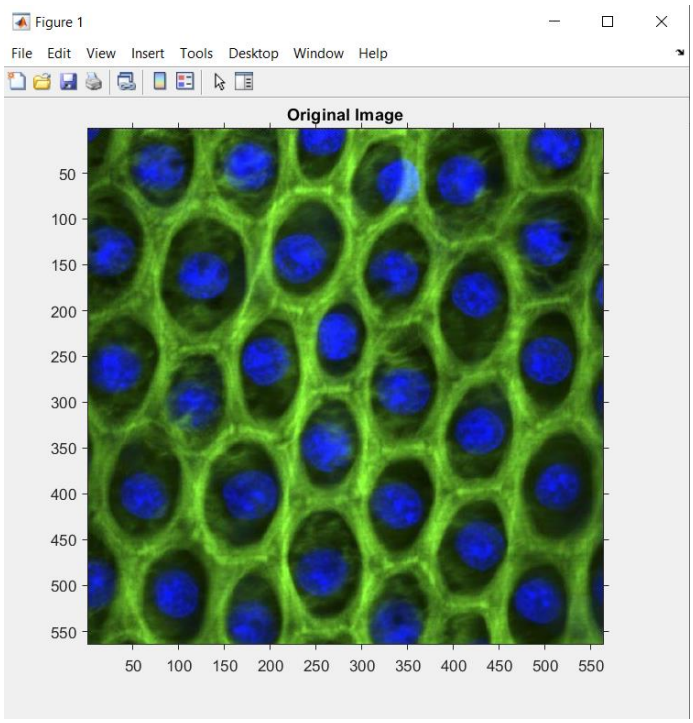

Figure.1 Original Image

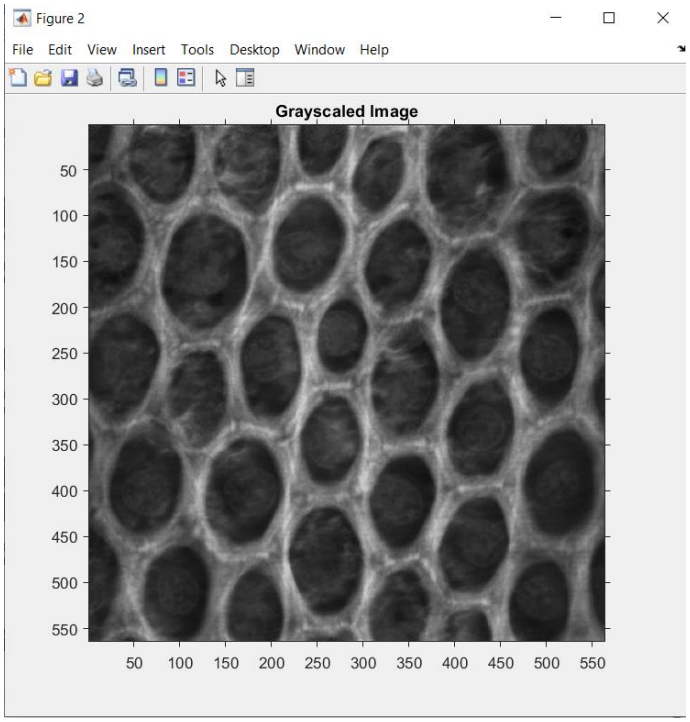

Figure.2 Grayscaled Image

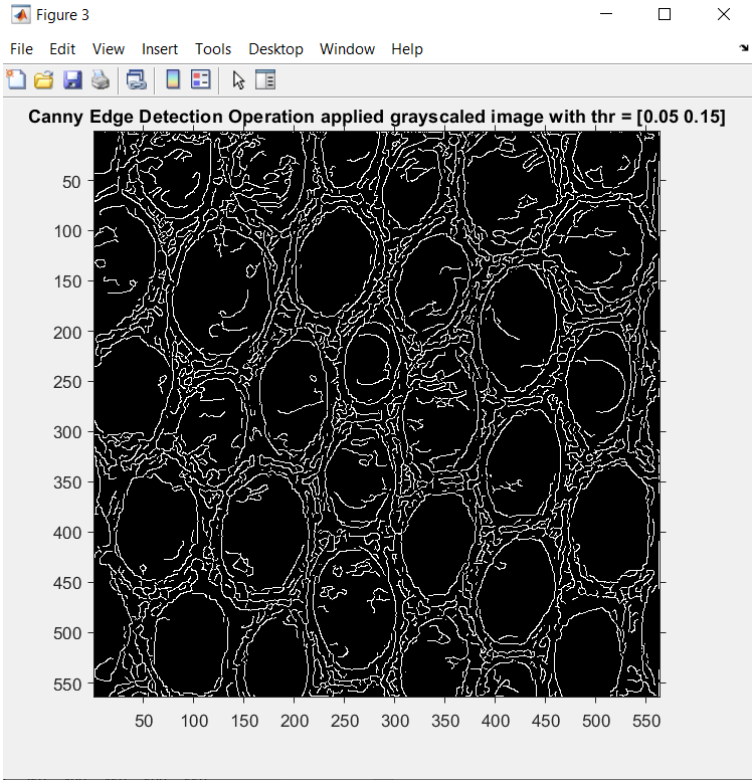

Figure.3 Canny Edge Detection Operation Applied (Tmin = 0.05, Tmax=0.15, sigma=1)

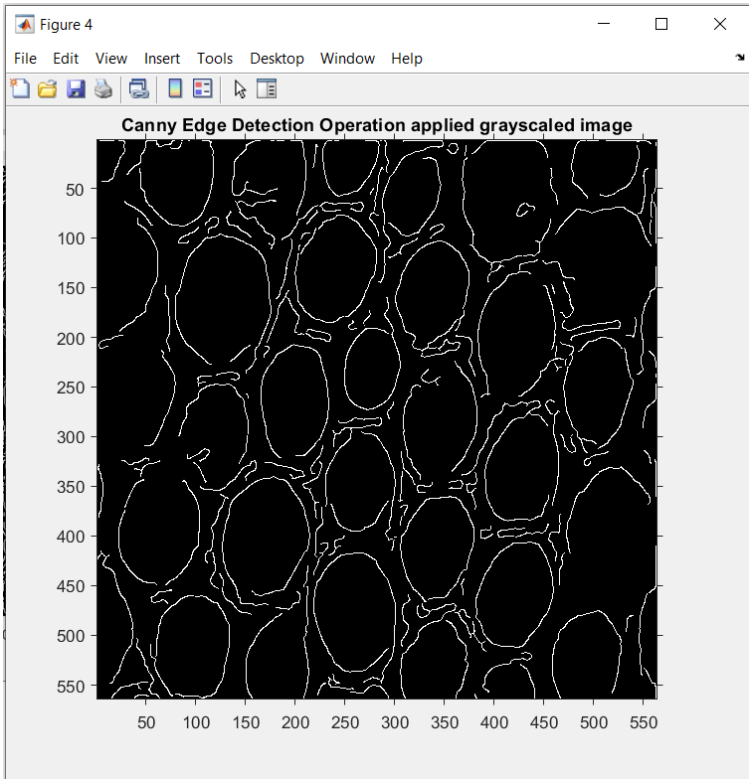

Figure.4 Canny Edge Detection Operation Applied

CODE PART

```
close all
clear all

%ASLI KURT _ 151220182068
%reading the image that named NEI-medialibrary-4506615.jpg
Image = imread("NEI-medialibrary-4506615.jpg");

%displaying the image that has read
figure;
imshow(Image);
axis on
title('Original Image');

%converting image to the graycale
Image_gray = rgb2gray(Image);

%displaying the grayscale image
figure;
imshow(Image_gray);
axis on
title('Grayscaled Image');

%definition of the parameters that used to apply canny edge
detection operation
Tlow = 0.05;
Thigh = 0.15;
sigma = 1;

%applying canny edge detecting operation
Image_canny = edge(Image_gray,'Canny',[Tlow Thigh], sigma);

%displaying the obtained image
figure;
imshow(Image_canny);
axis on
title('Canny Edge Detection Operation applied grayscaled image with
thr = [0.05 0.15]');

%definition of the parameter that used optimum sigma value in canny
operation
sigma_opt = 2;

%applying canny edge detecting operation
E_canny_opt = edge(Image_gray,'Canny',graythresh(Image_gray),
sigma_opt);
%[E_canny_opt,optThr] = edge(Image_gray,'canny');
```

```
%displaying the obtained image
figure;
imshow(E_canny_opt);
axis on
title('Canny Edge Detection Operation applied grayscaled image');
```

PART 2

The work done in this part is similar to the previous part.

First, the original image is read and displayed. Then, only the information in the Blue channel of the original image in RGB format was parsed. In other words, it is extracted because only the Blue channel will be used. And the resulting Blue channel is displayed.

Then, the parameters to be used for the Canny Edge Detection operation are defined. These parameters are the same as the parameters in the previous part. In other words, minimum threshold (Tlow_blue), maximum threshold (Thigh_blue) and sigma (sigma_blue) parameters are defined.

Canny Edge Detection operation has been applied and the result is displayed.

The same operation steps were repeated after calculating optimum values. In light of the results obtained from the repeated operation, the optimum sigma value was determined as '6'. Optimum threshold values were calculated with the graythresh function, which makes its calculations using Otsu's Method. The result obtained in this step is displayed as a figure.

Then, the circles in the image were detected using the Hough transform. This was completed using the imfindcircles function in the MATLAB program. The result obtained in this step is displayed as a figure.

Finally, in addition to the center coordinate and radius information of each circle calculated in the previous step, the circumference information calculated with the help of the circle circumference formula is displayed on the original image.

All images obtained throughout the study are included in the Part 2 in the form of figures, and the axis information of all images is included. Code blocks are available at the end of the part.

Implications

Using only the information of the blue channel of the original image is much more advantageous in terms of detecting cell nuclei because the blue color pixel density of cell nuclei

is much higher than the blue color density of the cell walls. This situation can be understood more clearly after extracting the blue channel of the original image. The fact that cell nuclei are more detectable than cell walls also provided an advantage during the Canny Edge Detection operation. Threshold and Sigma parameters (Tlow_blue=0.05, Thigh_blue=0.15 and sigma_blue=1) requested to be used in the assignment are not sufficient for the Part in question to achieve its purpose. Therefore, more appropriate threshold and sigma values were calculated. As a result of consecutive Canny Edge Detection operations, the optimum sigma value was calculated as '6'. This sigma value is the required parameter for the Gauss Filter. As a result of the calculation of these values, much purer and more accurate results were obtained compared to before. All cell nuclei were detected using the Hough Transform algorithm, and the detected cell nuclei were colored with a red circle. Using the center coordinates and radius information obtained from the imfindcircles function, the detected cell nuclei are shown on the original image.

As a result, it is much more accurate to perform operations via the blue channel extraction method to detect blue cell nuclei on the original image.

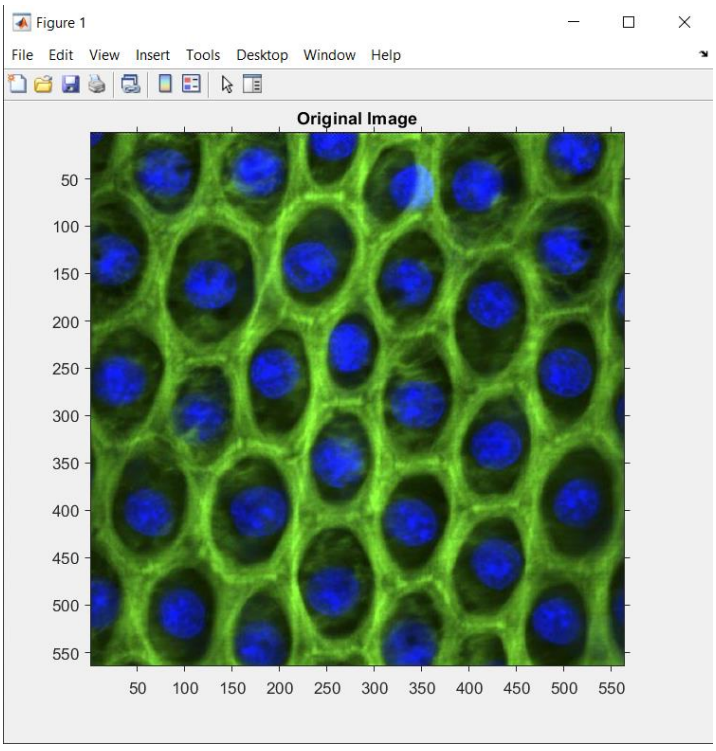

Figure.5 Original Image

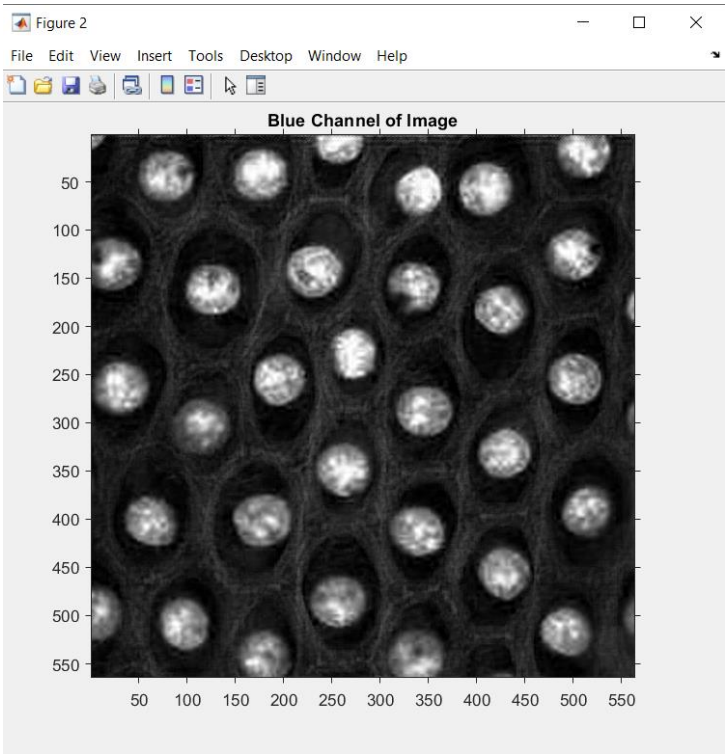

Figure.6 Blue Channel of the Original Image

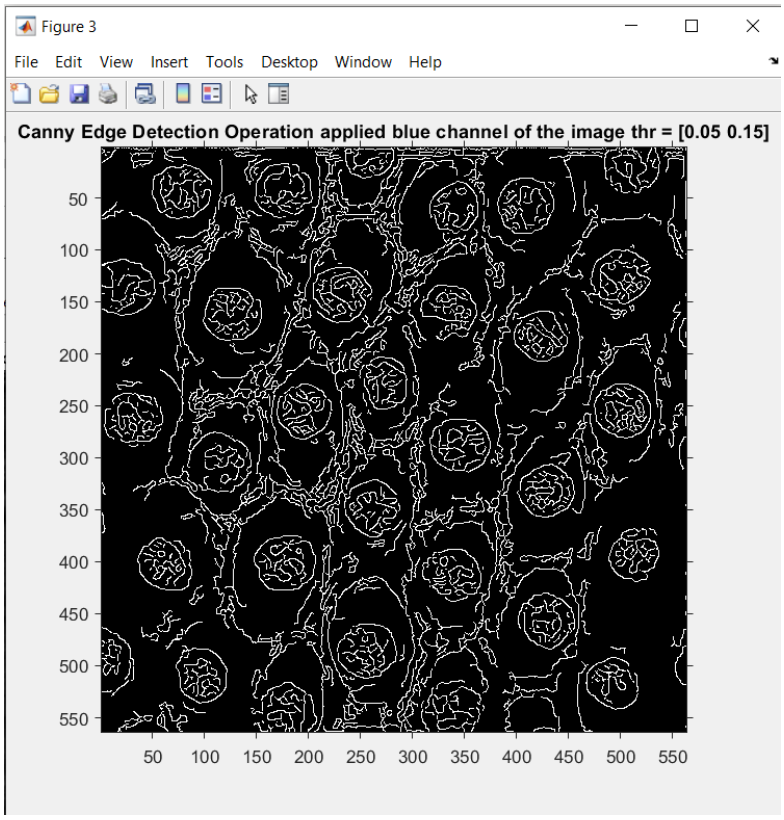

Figure.7 Canny Edge Detection Operation Applied (Tmin = 0.05, Tmax=0.15, sigma=1)

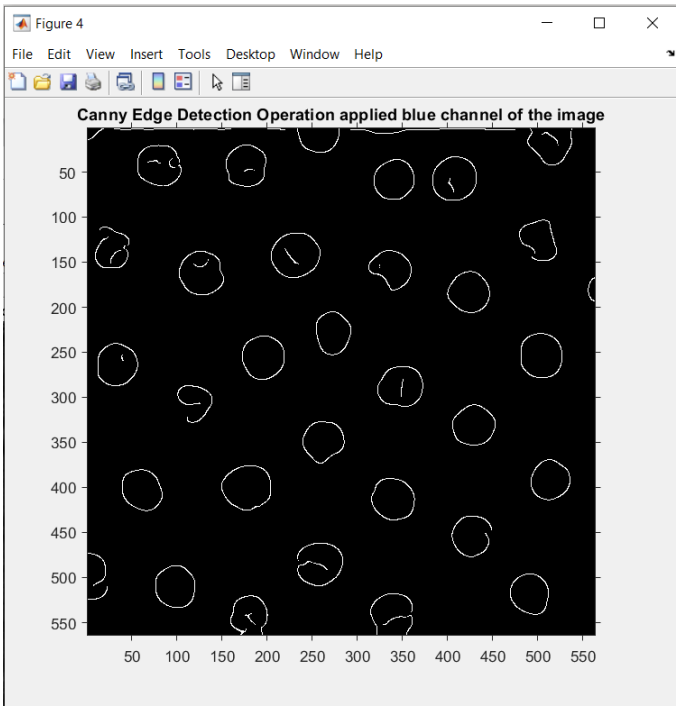

Figure.8 Canny Edge Detection Operation Applied

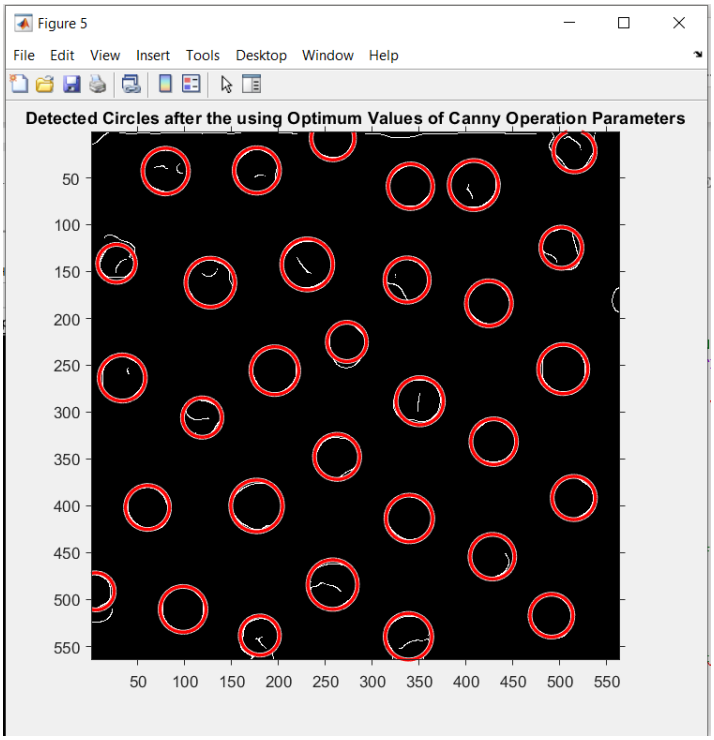

Figure.9 Hough Transformation Operation Applied

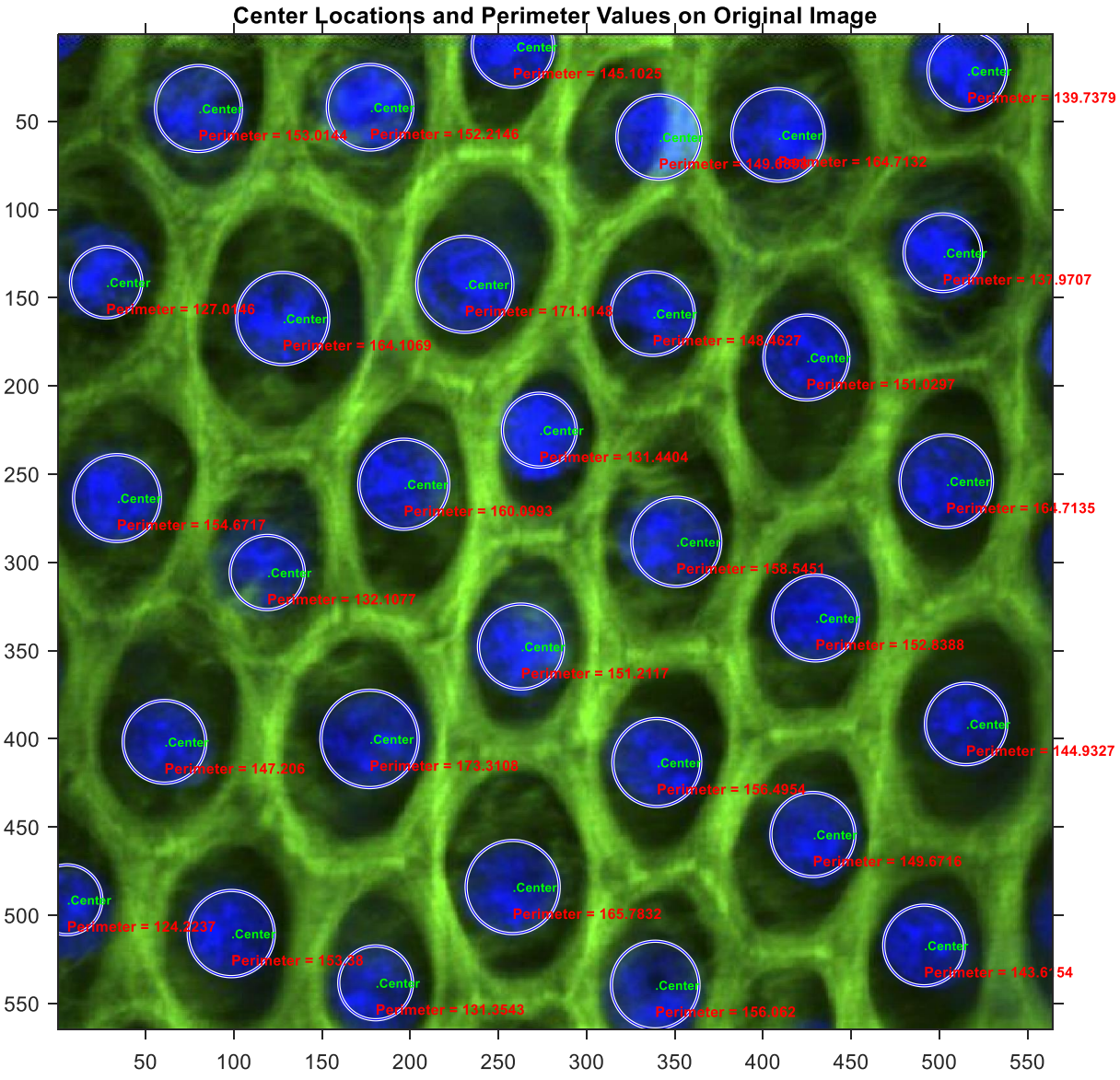

Figure.10 Center Locations and Perimeter Values on Original Image

CODE PART

```
close all
clear all

%ASLI KURT _ 151220182068
%reading the image that named NEI-medialibrary-4506615.jpg
Image = imread("NEI-medialibrary-4506615.jpg");

%displaying the image that has read
```

```
figure;
imshow(Image);
axis on
title('Original Image');

%extract the blue channel
Image_blue_channel = Image(:, :, 3); %the image is in RGB format

%displaying the blue channel of the image
figure;
imshow(Image_blue_channel);
axis on
title('Blue Channel of Image');

%definition of the parameters that used to apply canny edge
detection operation
Tlow_blue = 0.05;
Thigh_blue = 0.15;
sigma_blue = 1;

%applying canny edge detecting operation
%BW = edge(I,method,threshold,sigma) has used.
Image_canny_blue = edge(Image_blue_channel,'Canny',[Tlow_blue
Thigh_blue], sigma_blue);

%displaying the obtained image
figure;
imshow(Image_canny_blue);
axis on
title('Canny Edge Detection Operation applied blue channel of the
image thr = [0.05 0.15]');

%definition of the parameter that used optimum sigma value in canny
operation
sigma_blue_opt = 6;

%applying canny edge detecting operation
E_canny_blue_opt =
edge(Image_blue_channel,'Canny',graythresh(Image_blue_channel),
sigma_blue_opt);
%[E_canny_blue_opt,optThr_blue] = edge(Image_blue_channel,'canny');

%displaying the obtained image
figure;
imshow(E_canny_blue_opt);
axis on
title('Canny Edge Detection Operation applied blue channel of the
image');
```

```
%finding circles by using imfindcircles function
[centers, radii] = imfindcircles(E_canny_blue_opt, [14, 100]);
%radius has selected between 14 and 100

%display the original image with circles
figure;
imshow(E_canny_blue_opt);
axis on
title('Detected Circles after the using Optimum Values of Canny
Operation Parameters');

%draw circles on the image that obtained after applying canny edge
operation
viscircles(centers, radii, 'EdgeColor', 'r');

%displaying the image that has obtained
figure;
imshow(Image);
axis on
title('Center Locations and Perimeter Values on Original Image');
viscircles(centers, radii, 'EdgeColor', 'b' , 'LineWidth' , 0.5);

for i = 1:size(centers, 1)

    %defining the informations that center text, perimeter text and
value
    center = '.Center';
    perimeterText = 'Perimeter = ';
    perimeterValue = 2 * pi * radii(i); % perimeter formula =
2*pi*radius

    %put the parameters together
    perimeter = [perimeterText num2str(perimeterValue)];

    %display the center of the circles and perimeter values by using
text()
    text(centers(i, 1), centers(i, 2), center, 'Color', 'g',
'FontSize', 6, 'FontWeight', 'bold');
    text(centers(i, 1), centers(i, 2) + 15, perimeter, 'Color', 'r',
'FontSize', 7, 'FontWeight', 'bold');

end
```

PART 3

First, the original image is read and displayed. Then, the image defined in RGB space was converted using the HSV space rgb2hsv function. This transformation was made because the H, S and V channels belonging to the image would be used. Then the Hue channel of the image is extracted and displayed.

Then, the parameters required for the Canny Edge Detection operation were defined. These parameters are determined as minimum threshold(Tlow_hue), maximum threshold(Thigh_hue) and sigma(sigma_hue).

After the necessary parameter definition, the Canny Edge Detection operation was performed with the help of the edge function and is displayed.

Then, optimum threshold and sigma values were calculated. At this stage, the graythresh function was used, just like in the previous Parts.

Finally, the Hough Transform operation phase was started. At this stage, the circles in the image were detected by using the imfindcircles function. The results obtained are displayed on the original image.

All images obtained throughout the study are included in the Part 3 in the form of figures, and the axis information of all images is included. Code blocks are available at the end of the part.

Implications

Using only the data from the Hue channel of the original image made the cell nuclei much less obvious than the cell walls. Therefore, when the Canny Edge Detection operation is applied, detecting cell walls instead of cell nuclei is a much more expected result. Therefore, detecting blue fluorescently stained cell nuclei by using only the Hue channel information of the original image is a disadvantage. Therefore, the intended results could not be achieved. Instead of applying the steps applied in this Part using only the Hue channel, taking the Value channel information into account will provide much more appropriate results.

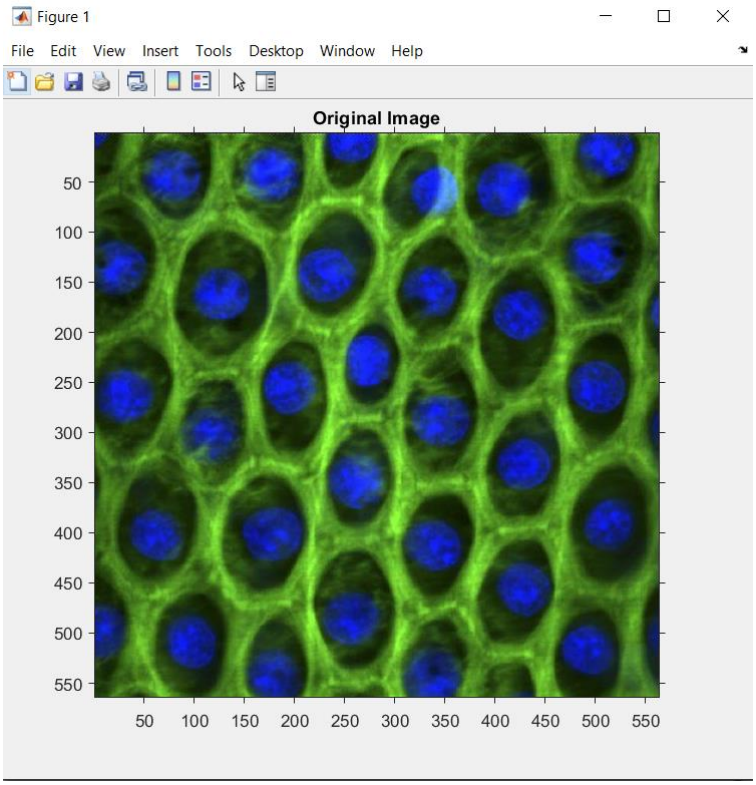

Figure.11 Original Image

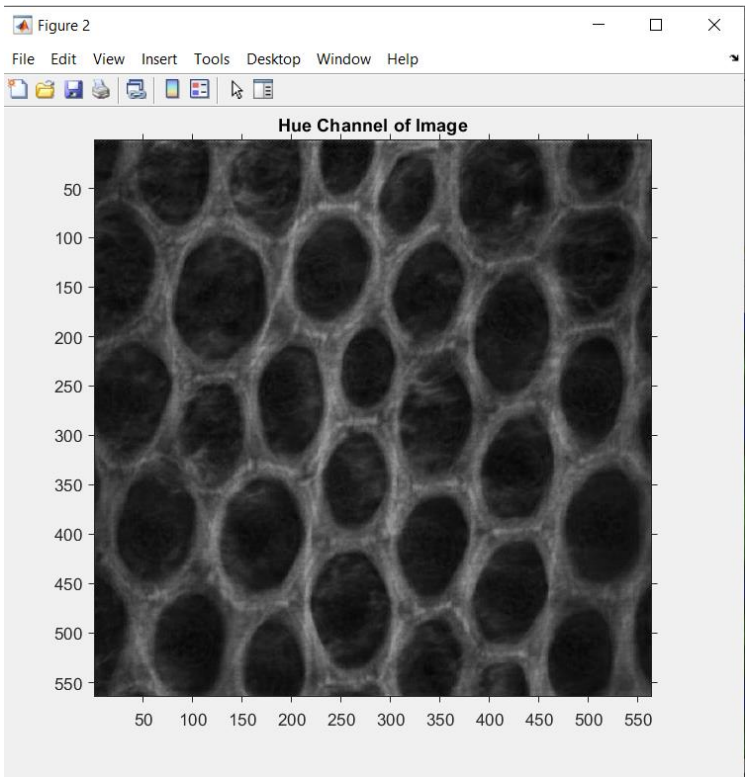

Figure.12 Hue Channel of the image

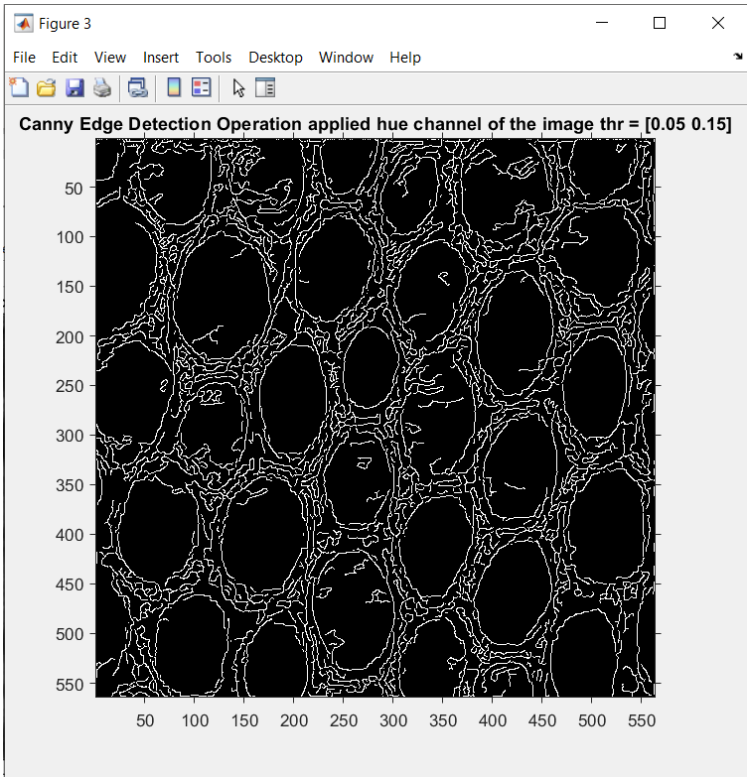

Figure.13 Canny Edge Detection Operation Applied (Tmin = 0.05, Tmax=0.15, sigma=1)

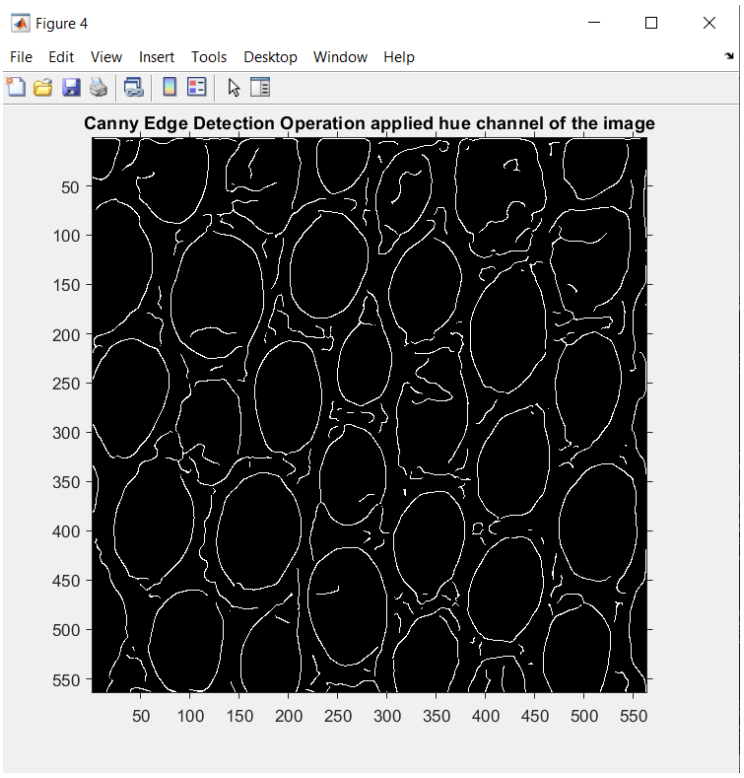

Figure.14 Canny Edge Detection Operation Applied

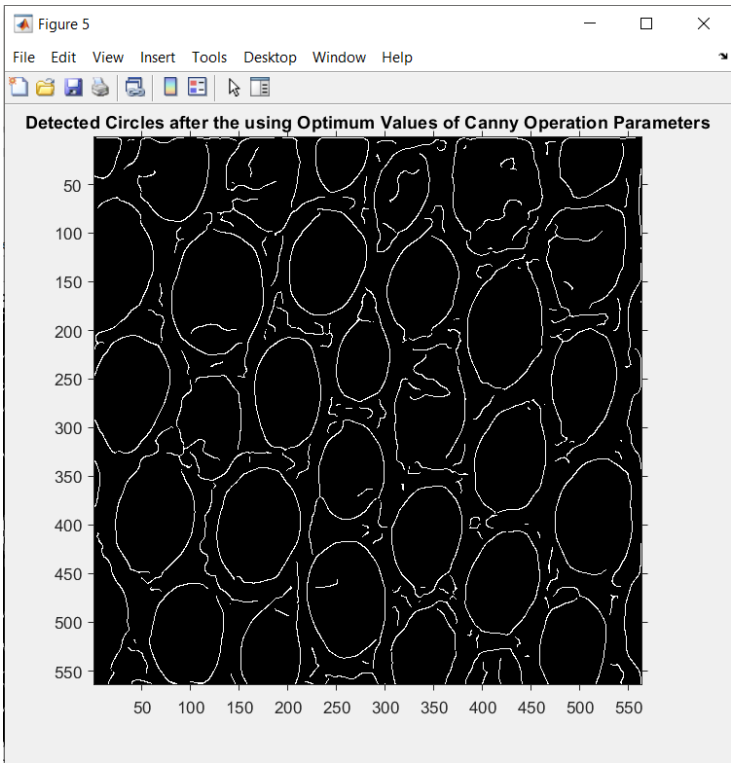

Figure.15 Hough Transform Operation Applied

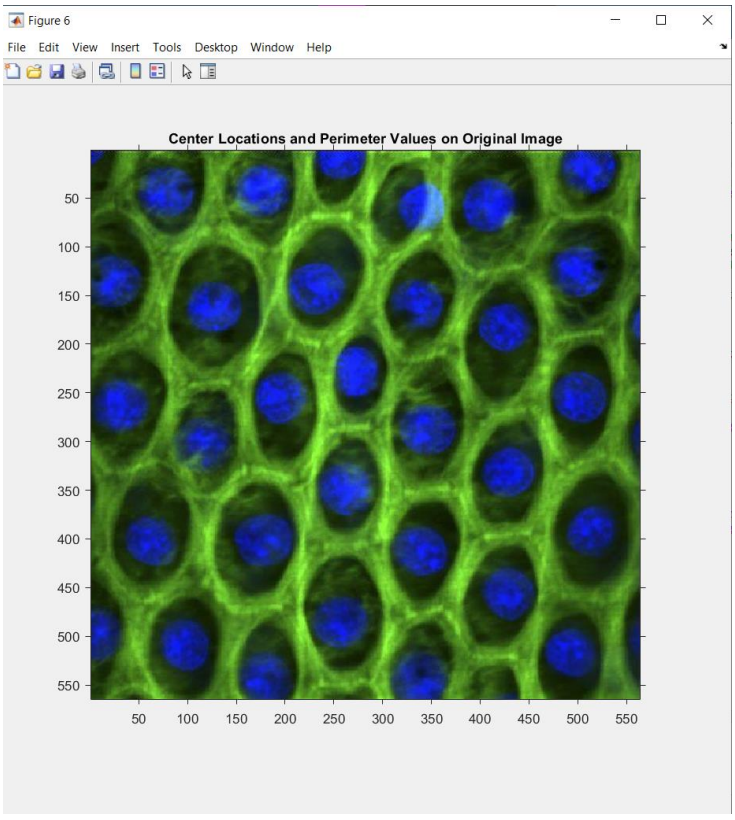

Figure.16 Center Locations and Perimeter Values on Original Image

INTRODUCTION TO IMAGE PROCESSING **Name:** Aslı Kurt

2023-2024 – FALL

CODE PART

```
close all
clear all

%ASLI KURT _ 151220182068
%reading the image that named NEI-medialibrary-4506615.jpg
Image = imread("NEI-medialibrary-4506615.jpg");

%displaying the image that has read
figure;
imshow(Image);
axis on
title('Original Image');

%converting image to the HSV space
Image_hsv = rgb2hsv(Image);

%extracting the hue channel
Image_hue_channel = Image(:, :, 1); %the image is in HSV format
%displaying the hue channel of the image
figure;
imshow(Image_hue_channel);
axis on
title('Hue Channel of Image');

%definition of the parameters that used to apply canny edge
detection operation
Tlow_hue = 0.05;
Thigh_hue = 0.15;
sigma_hue = 1;

%applying canny edge detecting operation
%BW = edge(I,method,threshold,sigma) has used.
Image_canny_hue = edge(Image_hue_channel,'Canny',[Tlow_hue
Thigh_hue], sigma_hue);

%displaying the obtained image
figure;
imshow(Image_canny_hue);
axis on
title('Canny Edge Detection Operation applied hue channel of the
image thr = [0.05 0.15]');

%definition of the parameter that used optimum sigma value in canny
operation
sigma_hue_opt = 4;

%applying canny edge detecting operation
```

```
E_canny_hue_opt =
edge(Image_hue_channel,'Canny',graythresh(Image_hue_channel),
sigma_hue_opt);
%[E_canny_hue_opt,optThr_hue] = edge(Image_hue_channel,'canny');

%displaying the obtained image
figure;
imshow(E_canny_hue_opt);
axis on
title('Canny Edge Detection Operation applied hue channel of the
image');

%finding circles by using imfindcircles function
[centers, radii] = imfindcircles(E_canny_hue_opt, [14, 100]);
%radius has selected between 14 and 100

%display the original image with circles
figure;
imshow(E_canny_hue_opt);
axis on
title('Detected Circles after the using Optimum Values of Canny
Operation Parameters');

%draw circles on the image that obtained after applying canny edge
operation
viscircles(centers, radii, 'EdgeColor', 'r');

%displaying the image that has obtained
figure;
imshow(Image);
axis on
title('Center Locations and Perimeter Values on Original Image');
viscircles(centers, radii, 'EdgeColor', 'b' , 'LineWidth' , 0.5);

for i = 1:size(centers, 1)

    %defining the informations that center text, perimeter text and
value
    center = '.Center';
    perimeterText = 'Perimeter = ';
    perimeterValue = 2 * pi * radii(i); % perimeter formula =
2*pi*radius

    %put the parameters together
    perimeter = [perimeterText num2str(perimeterValue)];

    %display the center of the circles and perimeter values
    text(centers(i, 1), centers(i, 2), center, 'Color', 'g',
'FontSize', 6, 'FontWeight', 'bold');
```

```
    text(centers(i, 1), centers(i, 2) + 15, perimeter, 'Color', 'r',
'FontSize', 7, 'FontWeight', 'bold');

end
```