INFORMATION TECHNOLOGY UNIVERSITY

Report (Assignment 1)

Submitted To: Dr. Mohsen

Submitted By: Sheeza Shabbir

Subject: Computer Vision

Roll No: MSDS20028

Implementation

1. Generation of Masks

- To generate Mask for the images first I created the function Calculate_filter_size (T, Sigma) that gives the size we are going to use. By changing the value of sigma one can change the size of the filter. For experiments, I am using three values of sigma 0.5,1, and 2.

sigma	Filter Size
0.5	(3,3)
1	(5,5)
2	(7,7)

While keeping T = 0.3 the same for all sigmas. Later we will see that how changing the size of the filter has a significant effect on output.

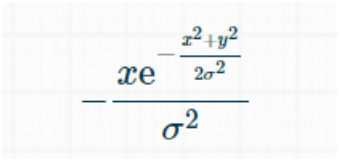
- The filter size is used in function Generate_Mask () to generate masks. In the Generate_Mask () I am using range() to define the range of values for x and y and np.meshgrid() is creating a rectangular grid of x and y values.
- To apply a Gaussian filter for smoothing Image the function I am using is apply_Gaussian(X, Y, sigma) where X and Y are the masks we get from the Generate_mask(X, Y,sHalf) and sigma is the value we set for the filter size. Gaussian Formula:

$$e^{-\frac{x^2+y^2}{2\sigma^2}}$$

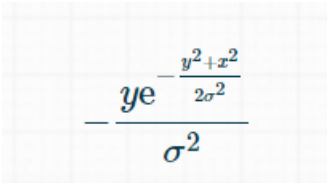
One thing we need to be careful about it is that when we use this equation in the code any change in the sign of the equation will affect the results. It will give you a different output.

- Taking the first derivative of Gaussian w.r.t x and y will result in Gx and Gy filter that will be scaled up using scaledown_Gradients() in which we multiply Gx and Gy with 255 and round it up to avoid floating-point values in the filter. The formula for the first derivative is given below:

Gx



Gy



Be careful with signs while using this in code. Changes in the signs will affect your results. Changing the sign will change your filter and the consequences of it are wrong results. It took me a lot of time to find this mistake in the code and tackle it.

2. Apply mask to the images:

Three steps we need to follow:

Grayscale Conversion:

First, we need to convert the image into grayscale if it has three channels. The reason behind converting it into the grayscale is that grayscale simplifies the algorithm and reduces computational costs. I have set a condition in apply_mask(img, filter) if the image is already in grayscale do not need to convert it otherwise change the dimensions.

Convert_grayscale(img): Passing image as input parameter

Smoothing the image:

After converting the image into Grayscale we need to smooth the image. Why do we need to do this? To avoid the noise affecting detection. What happens is that images contain noise in them this can be due to different reasons and when we apply first and second-order derivative filters, they are very sensitive to noise. SO, therefore, we use a Gaussian filter to remove this noise and later apply the mask on it. Without destroying the true edges Smoothing removes as much noise as possible.

ImageSmoothing(Grayscaleimg, Gaussain_mask) Passing Grayscale Image and mask as input parameters

Masking the Image:

Now convolving the image with the input mask gx and gy. And in the results, we get fx and fy. One thing over here while convolving the image we need to pad the borders of the image. Why do we do this? Sometimes filter does not perfectly fit the input image in the results what happens the information on the borders of images is not preserved as well as the information in the middle. To avoid this problem we use

padding. I am using cv2.copyMakeBorder(img,pad,pad,pad,pad,cv2.BORDER_REPLICATE) it will pad the image that will be the replication of original edges. Finally, if your filters are fine then this function will give the right results.

3. Computing Gradients Magnitude:

Magnitude_of_Gradient(fx,fy) for computing Gradient magnitude. The gradient magnitude is the scalar quantity that tells us about the local rate of change in the scaler field. Parameters I am passing are the results from apply_mask(img, mask) where masks are gx and gy.M is computed over here using the magnitude formula given below.

$$M=\sqrt{f_x^2 + f_y^2}$$

To write output to the image files, the min and max values are reduced 0 to 255 respectively using scaledown_Gradients (M) where M is passed as a parameter.

4. Computing Gradients Direction:

Gradients_Direction (fx,fy) for computing Gradient direction. Gradient direction helps to find the edges as gradient direction is always perpendicular to the edge detection.

$$\theta = \arctan \frac{f_y}{f_x}$$

To compute this theta I am using np.arctan2 (fy,fx). The function is used for 2d operations. There are also other functions available some of them are for scalar values. But this function gives fine results so far. Next thing to round up the theta to min 0 and max 360 I am using np.rad2deg() and adding 180. We can do this manually by writing equations. But I got fine results from this method.

5. Non-Maximum-Suppression:

Quantization:

For this purpose using Quantization (D) function passing Direction as a parameter. Direction D we computed from the function Gradient_Direction(fx,fy). Why we are doing Quantization? For non-maximum suppression in a given pixel, there are 8 neighbors. There are four possible directions, 0, 45,90, and 135 degrees. The angles are being quantized into these 4 angles using the Quantization function. We need to be careful while working on this any > or <= change in these signs can affect the final results.

Color Quantization:

Color_Quantization(Q,M) passing Quantization matrix and Magnitude. What this function is doing is assigning the color for each bin of the quantization matrix. Where the Magnitude is zero the color distribution over there will be zero otherwise 255. The function is just Quantization in colors.

Non-Maximum-Suppression:

def Non_Max_Suppression(Q, D, M) passing Quantization matrix, Direction, and Magnitude as parameters. To understand this function consider an image and let assume that the angle of the gradient at that point is 0 degrees. for which we are using Q[x,y] to check angle, the direction of the edge should be perpendicular to it. We will check for the left and right neighboring pixels values for this we are using M[x,y] that is magnitude. If the gradient at the present pixels is greater than its left and right pixel then it is considered to be an edge and save to the MS[x,y] otherwise it is discarded. Non-Maximum-Suppression gives us thin edges.

6. Hysteresis Thresholding:

def Hystereisis_Thresholding(MS,min_TH,max_TH) passing MS results from Non-Maximum-Suppression() function, Minimum threshold, and Maximum Threshold so far we have set. This is the final step I have tested the images with two different pairs of min_th and max_th which are (70,100) and (50,70) for different values of sigma. A lot of difference can be seen from the resultant images of both pairs. That I will discuss in the experiments given below. How it's working. Given the matrix of MS[x,y] (Non-Maximum-Suppression output) it checks the values if the pixel value is less than min then set new matrix TSD[x,y]=0, if the value is between min and max threshold set the TSD[x,y] =128 and otherwise set it to 255. Through TSD we are getting both weak and strong edges. In the next and final step tracking edge by hysteresis converting weak edges into strong ones if and only if one the pixel around being processed is the strong one.

Experiments

Experiment 1: Sigma = 0.5, T =0.3,filter (3,3)

LT=50 and HT=70

Image: Circle

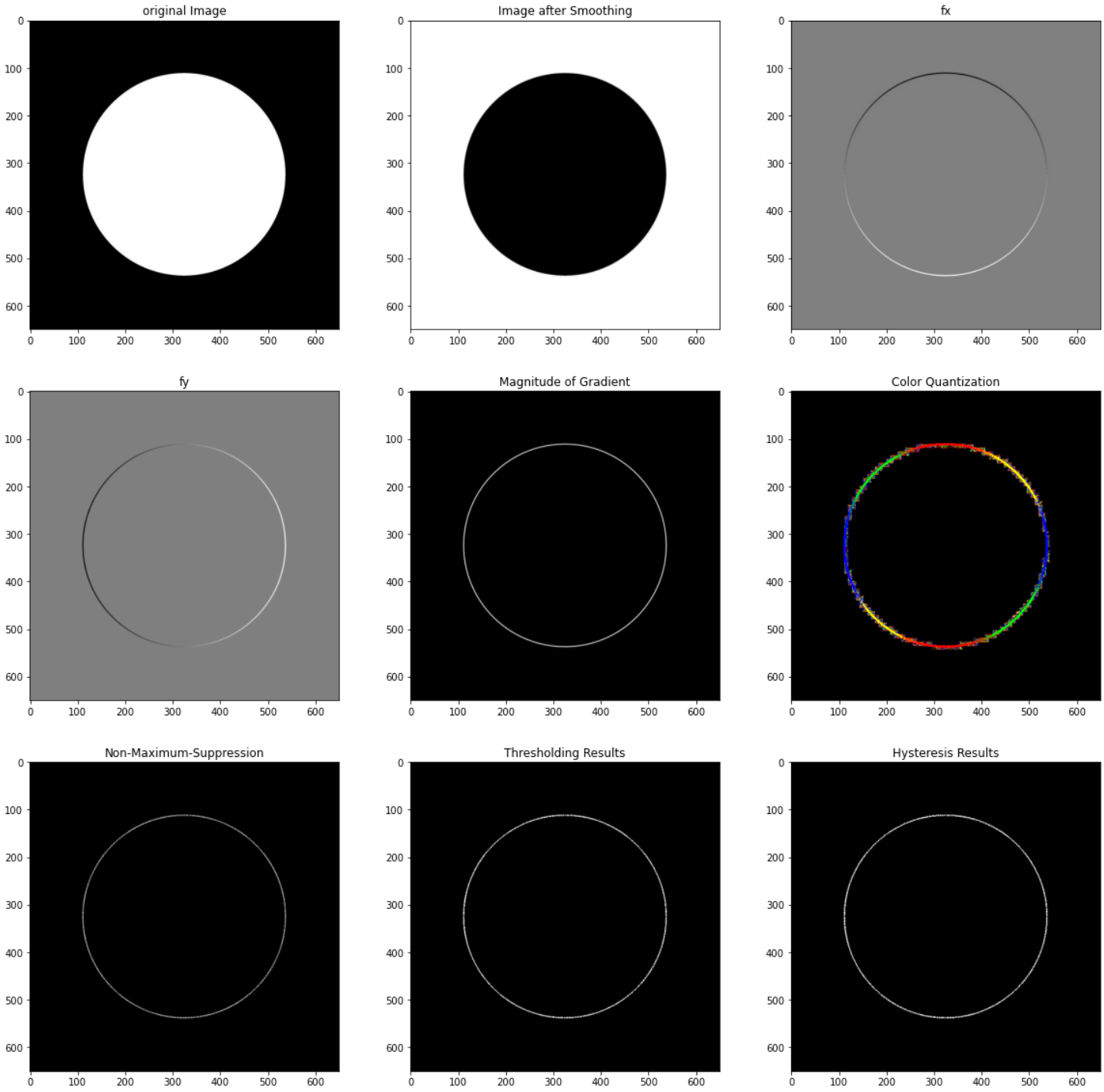


Image: X-ray

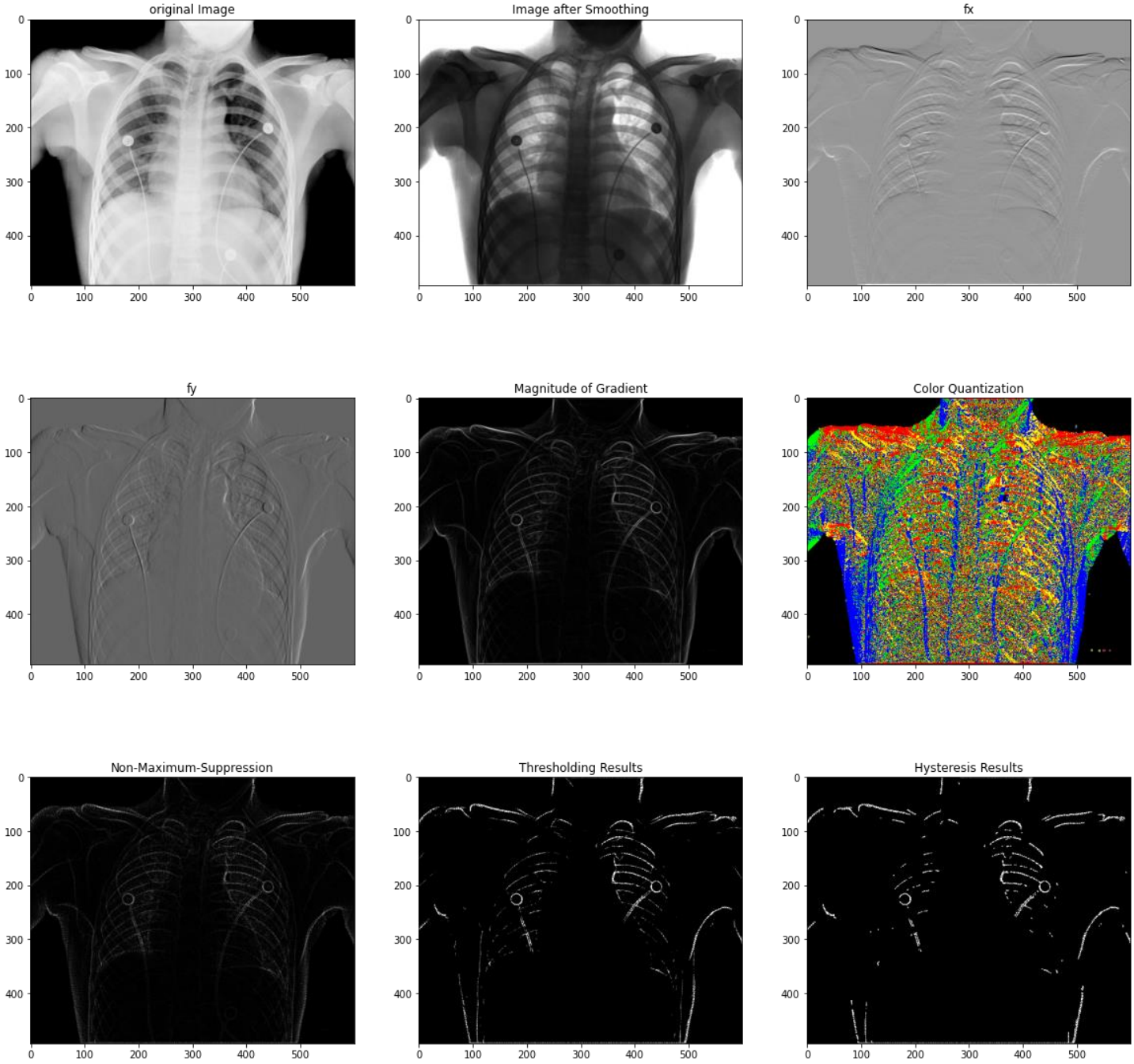

original Image
Image after Smoothing
fx
fy
Magnitude of Gradient
Color Quantization
Non-Maximum-Suppression
Thresholding Results
Hysteresis Results

Image: CT scan

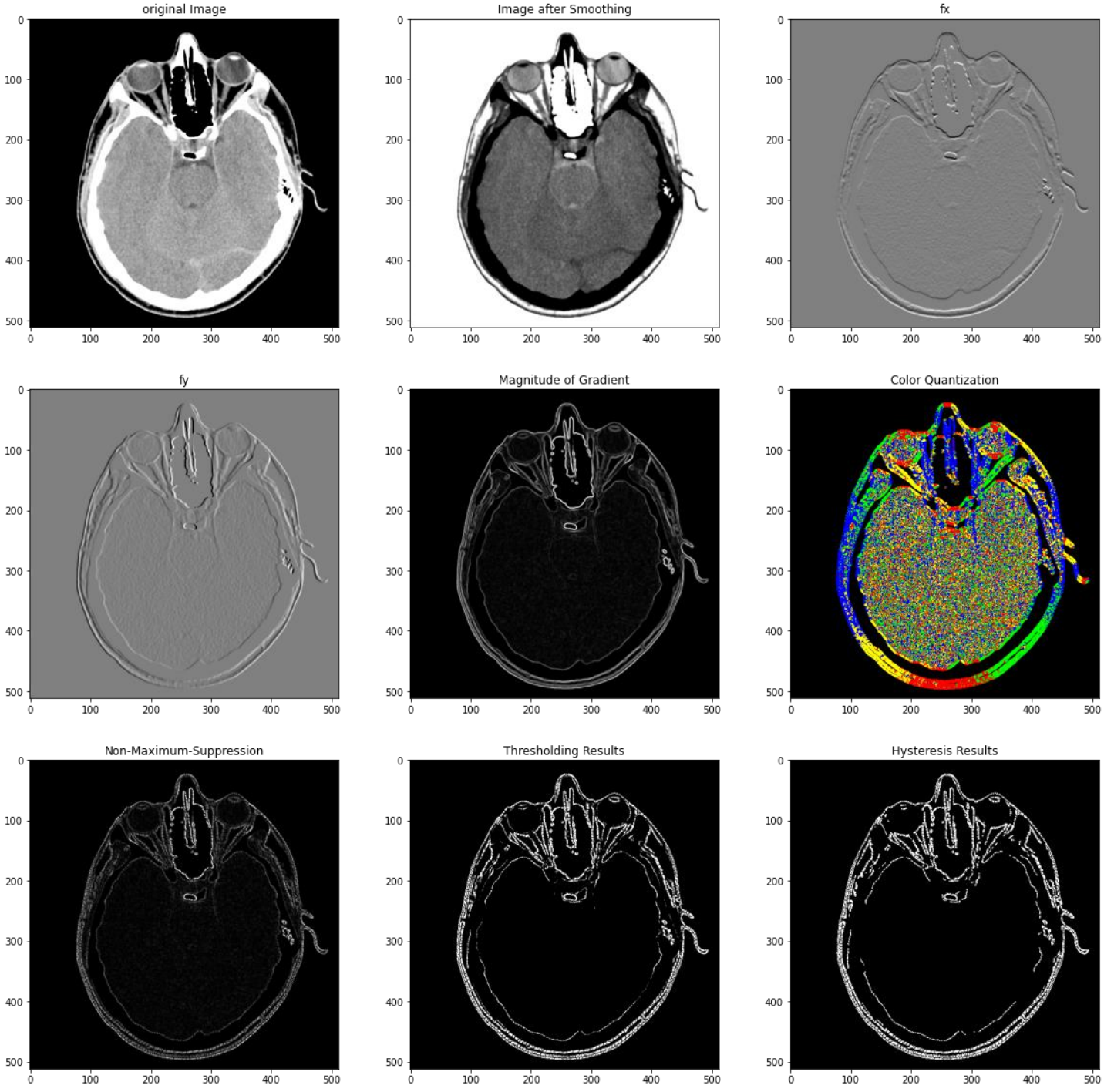


Image: mecca

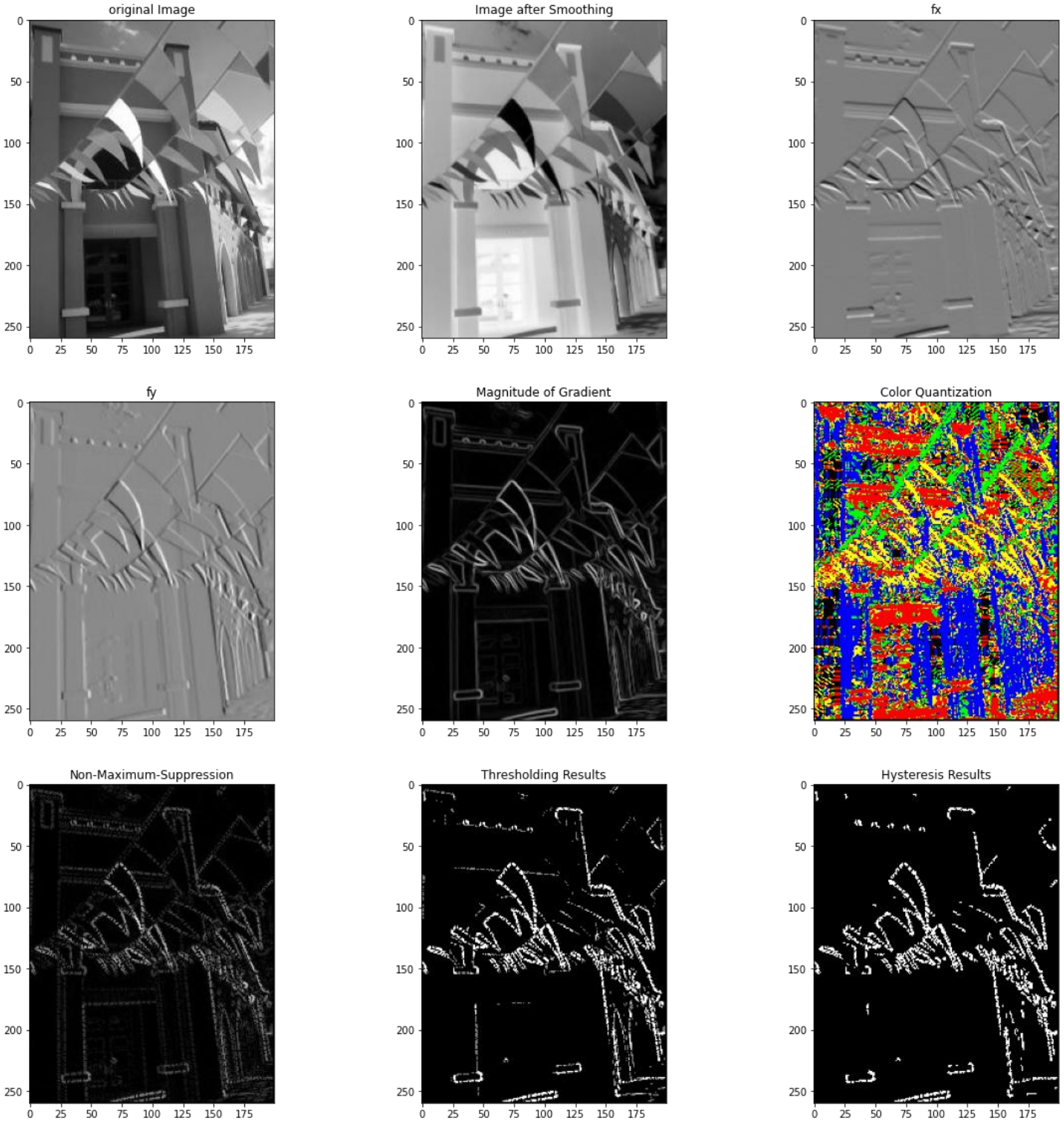
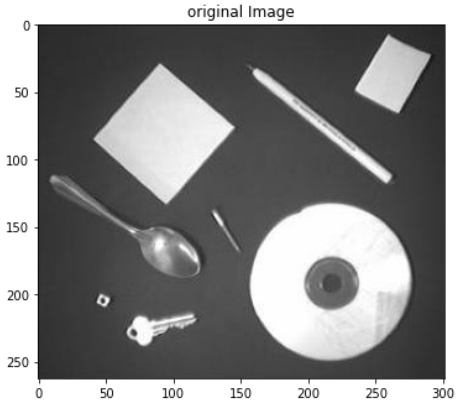

original Image
Image after Smoothing
fx
fy
Magnitude of Gradient
Color Quantization
Non-Maximum-Suppression
Thresholding Results
Hysteresis Results

Image: Shape


original Image

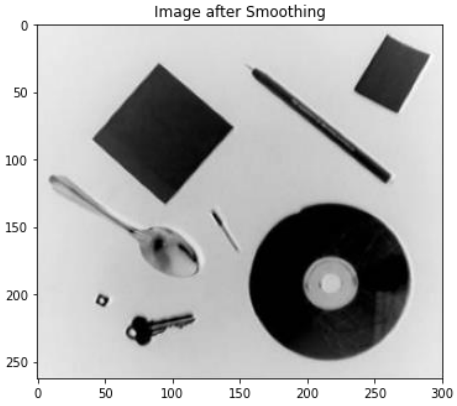
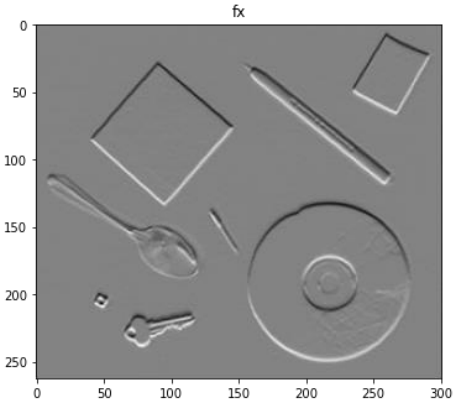
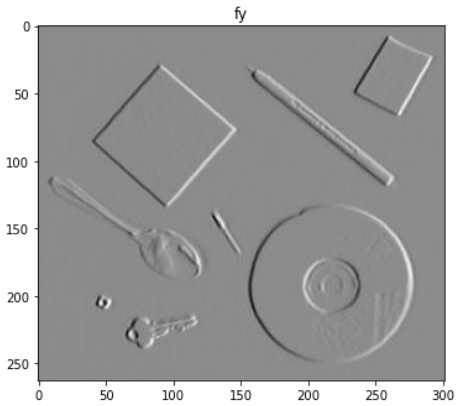
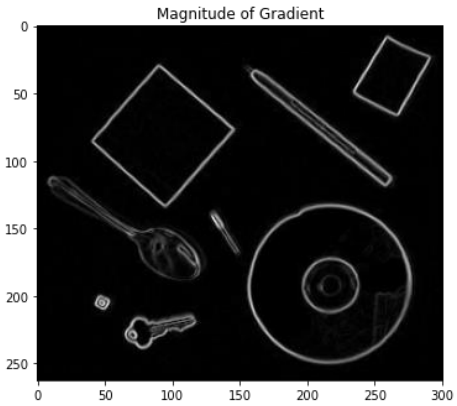
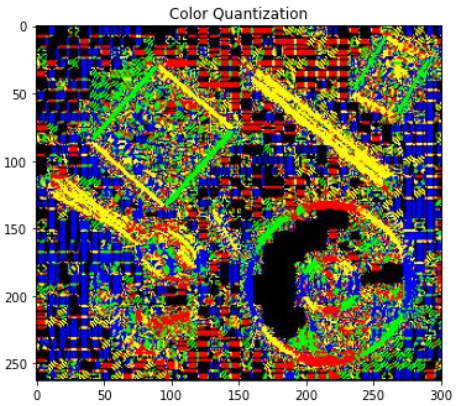


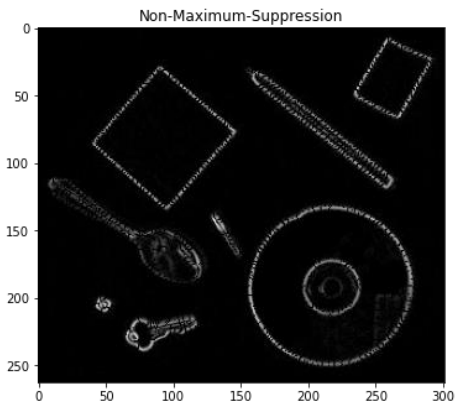
Image after Smoothing

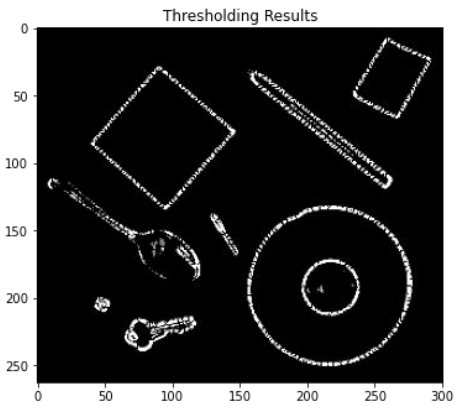

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

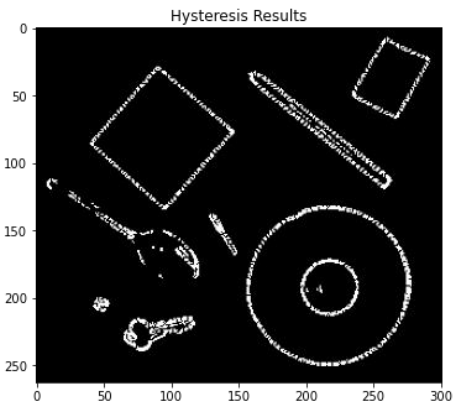
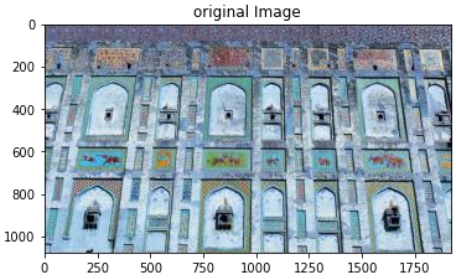
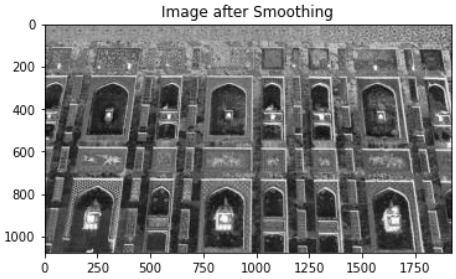
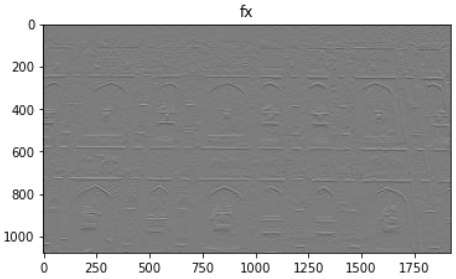
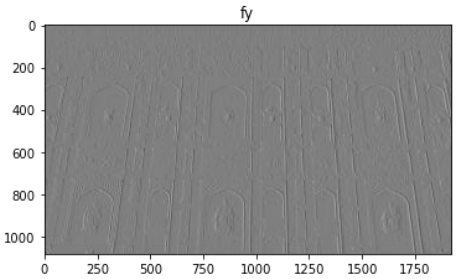

Hysteresis Results

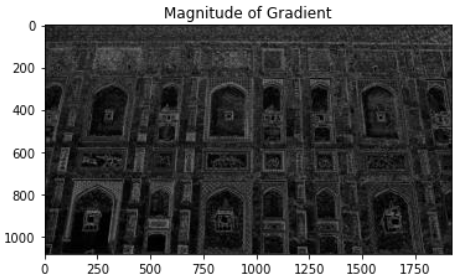
Image: wall Lahore

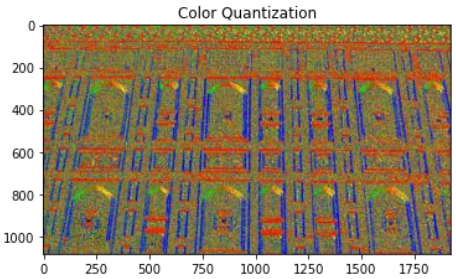


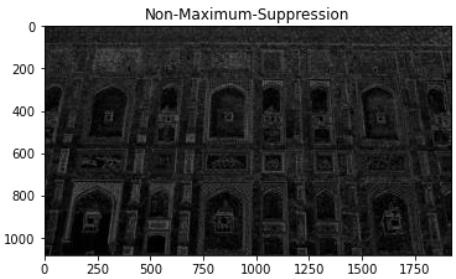


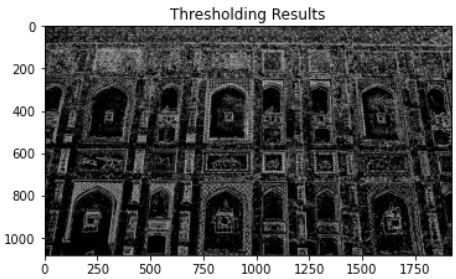


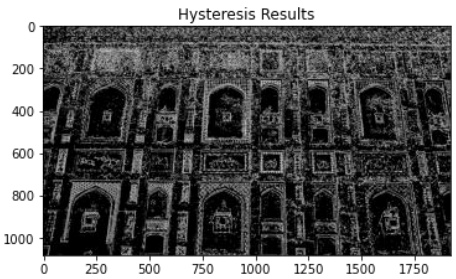












Experiment 2: Sigma = 1, T =0.3 and filter = (3, 3)

LT= 50 and HT= 70

Image: Circle

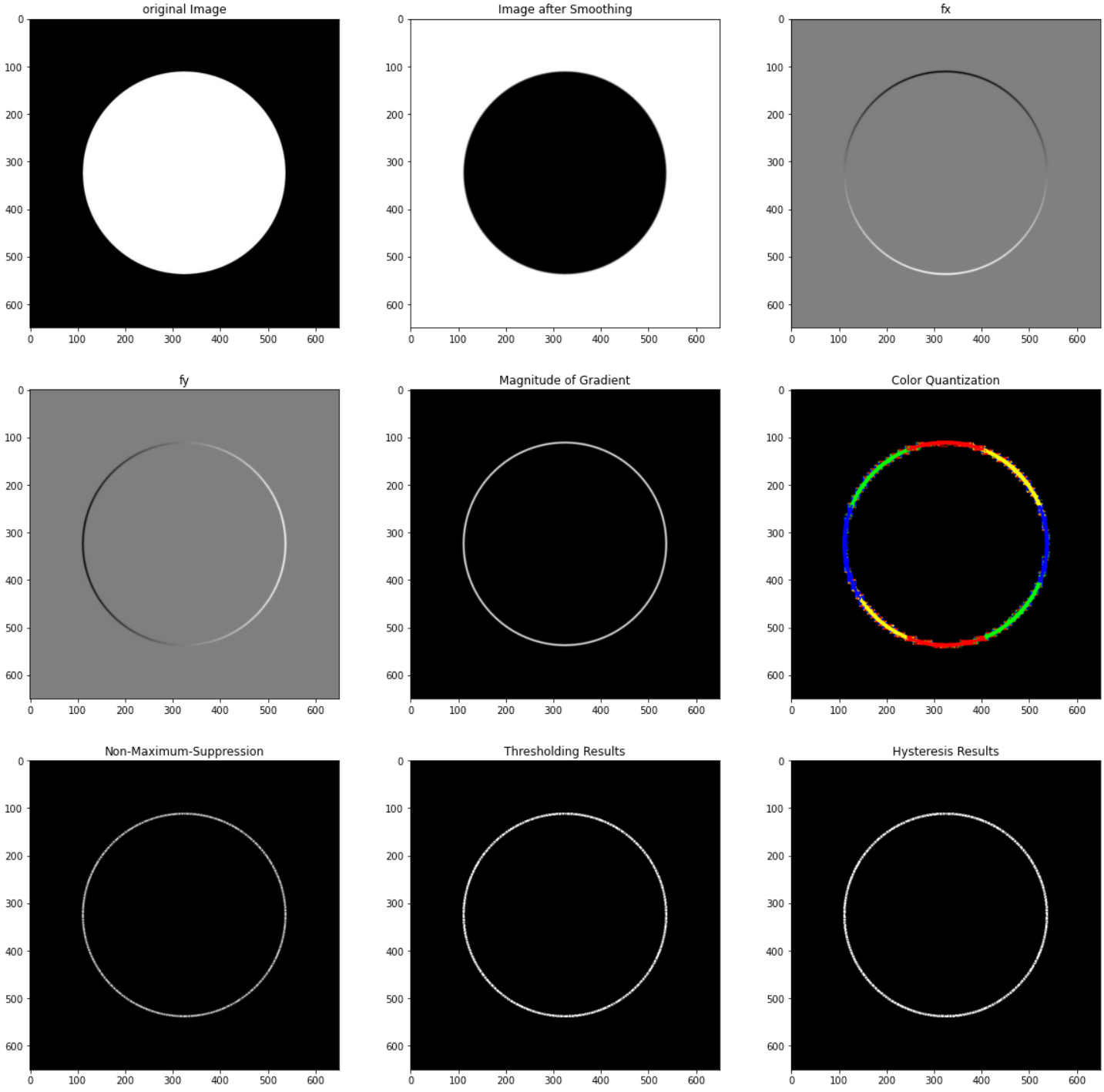
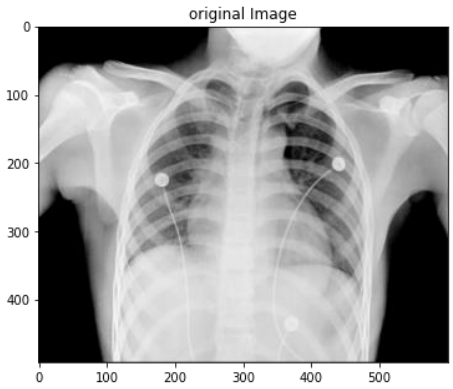


Image: x-ray


original Image

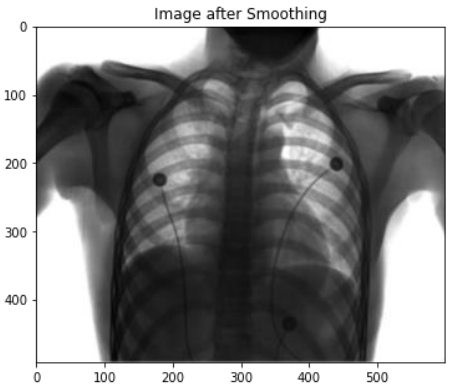
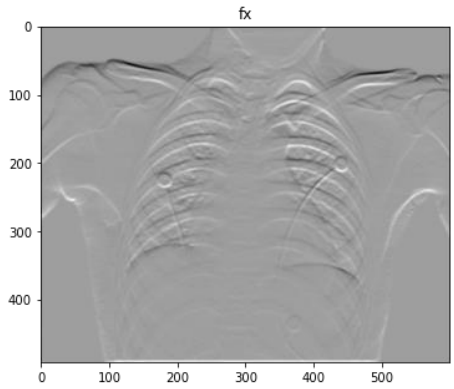
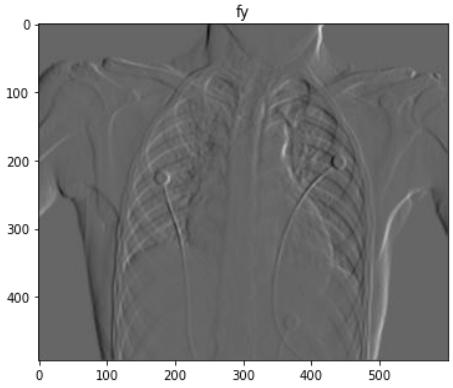
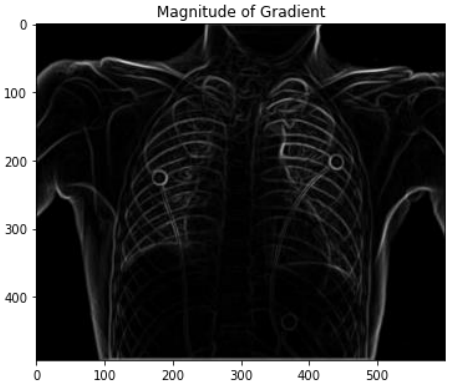
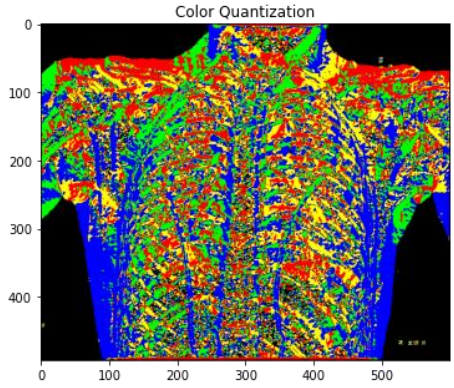


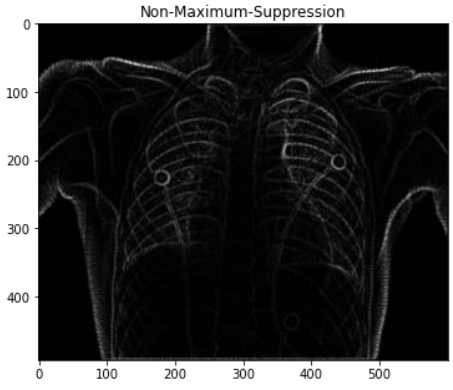
Image after Smoothing

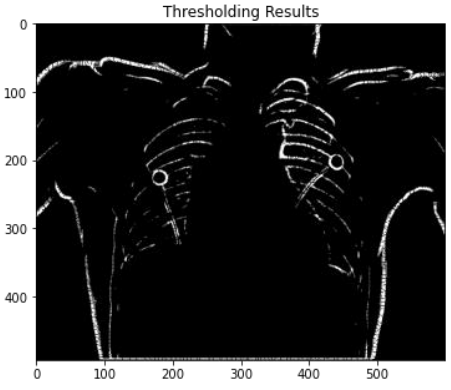

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

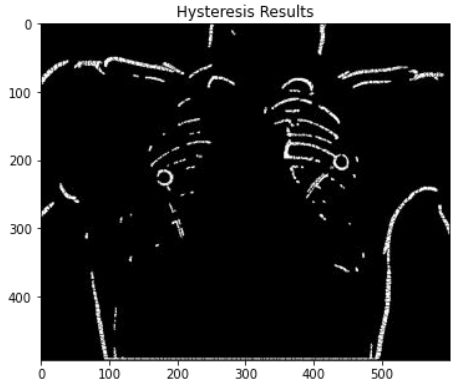

Hysteresis Results

Image: CT scan

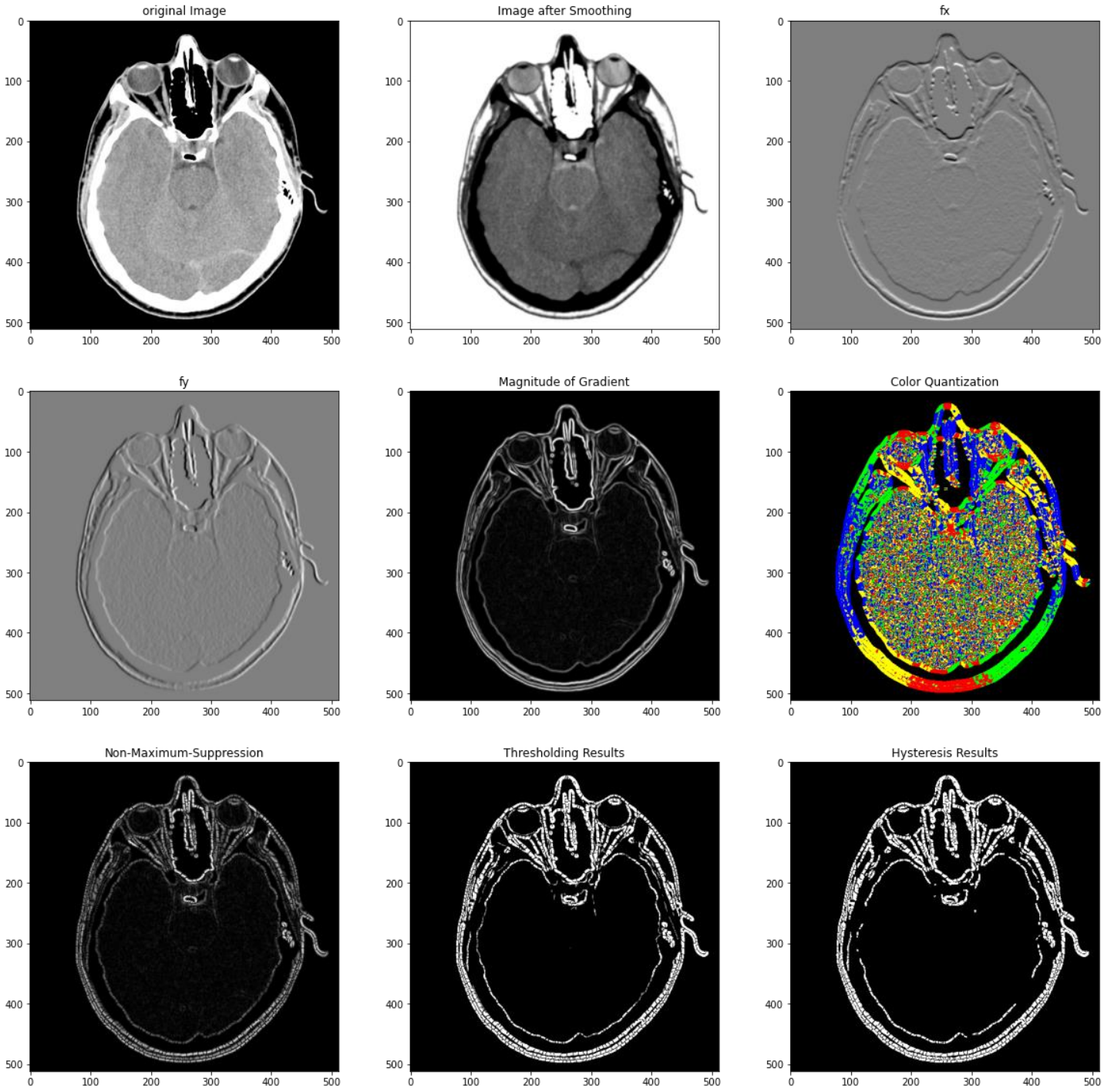
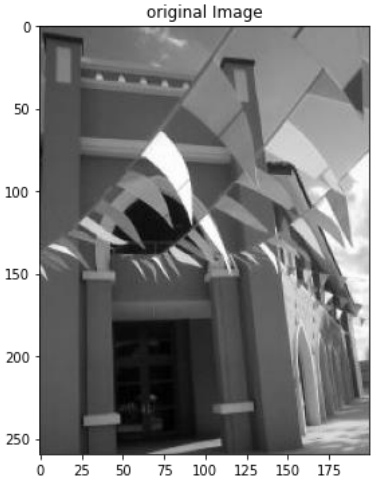


Image: mecca


original Image

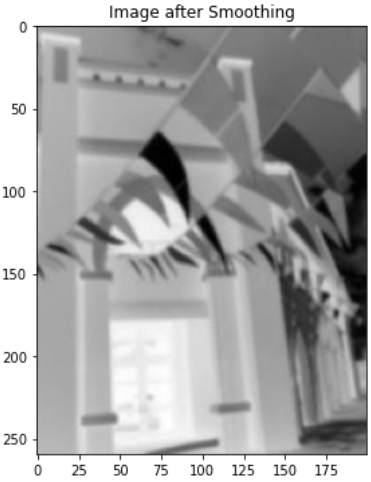
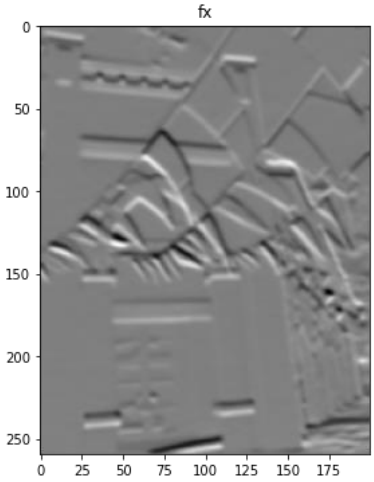
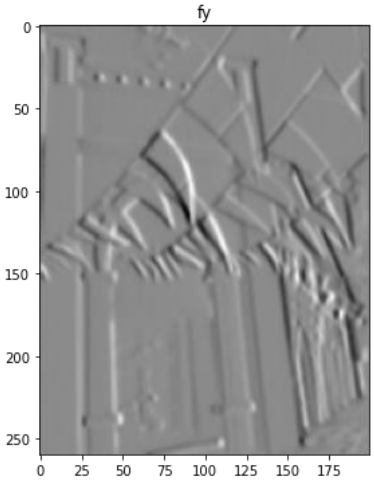
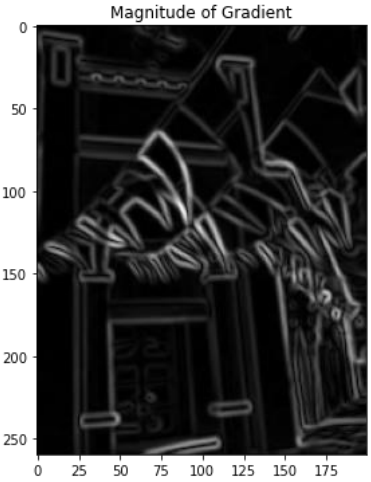
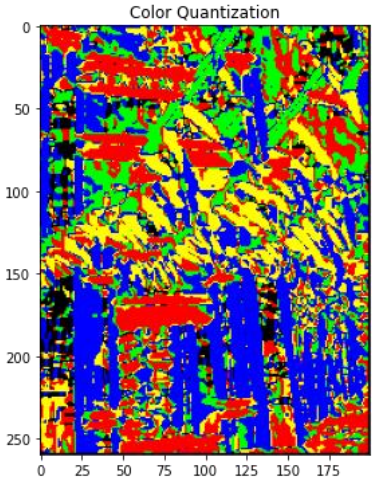


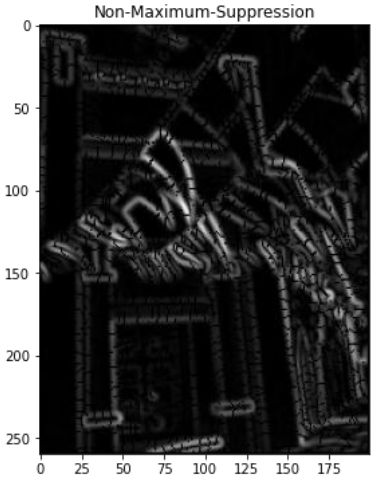
Image after Smoothing

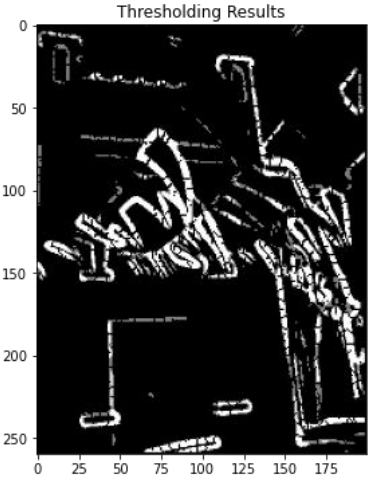

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

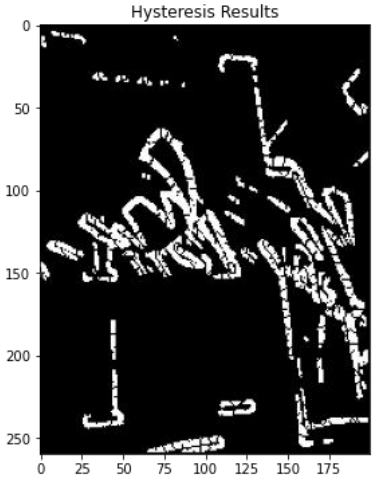
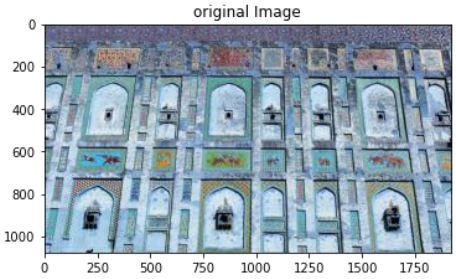
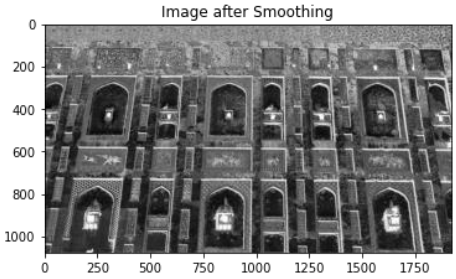
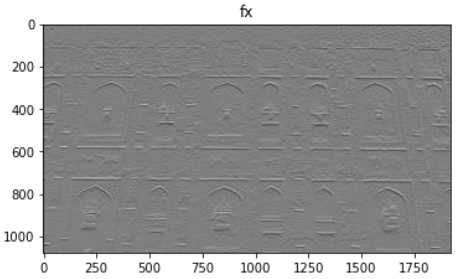
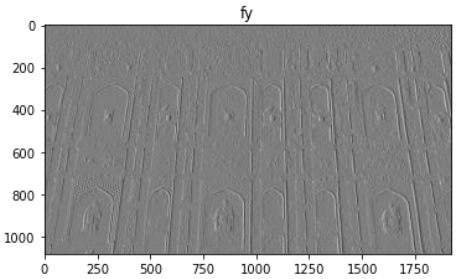

Hysteresis Results

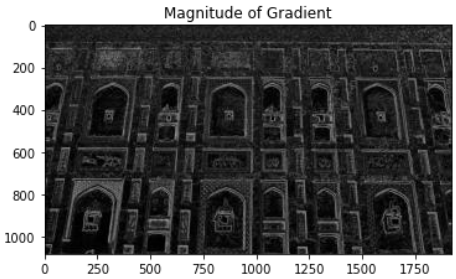
Image: Wall Lahore

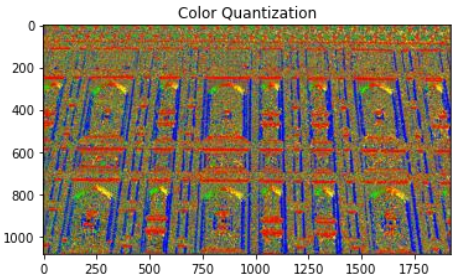


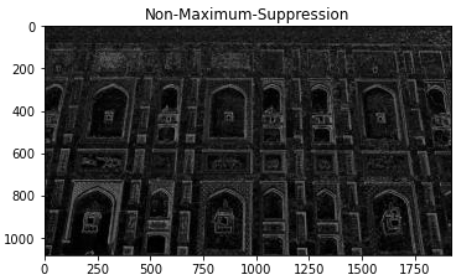


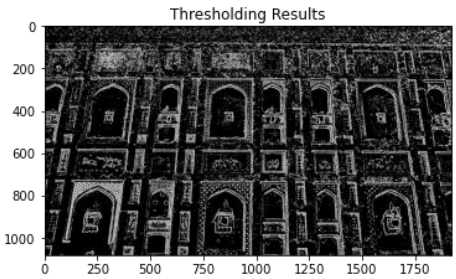












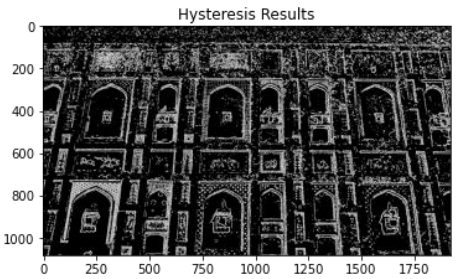
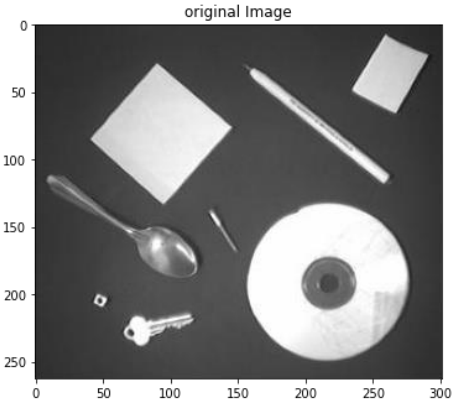


Image: Shape


original Image

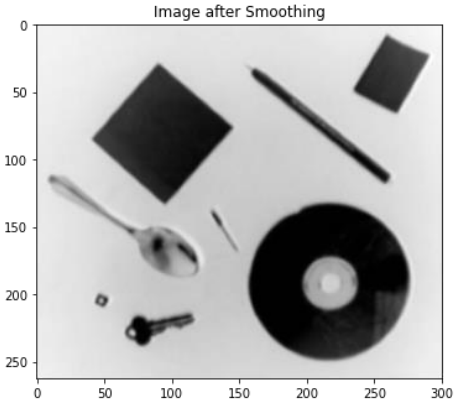
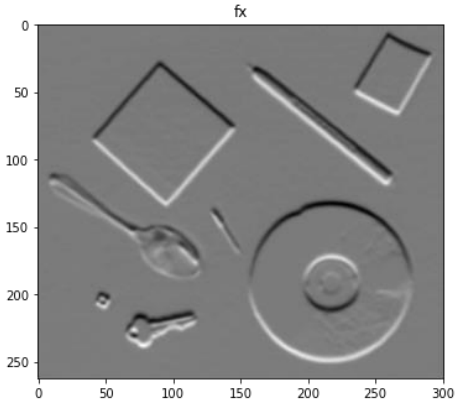
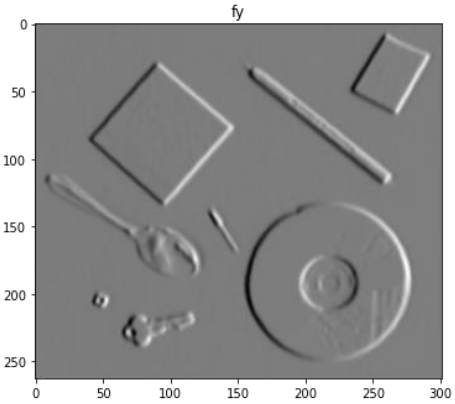
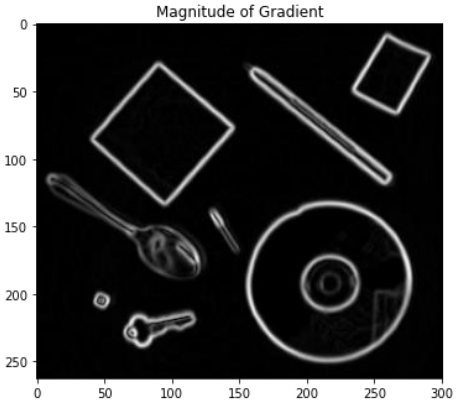
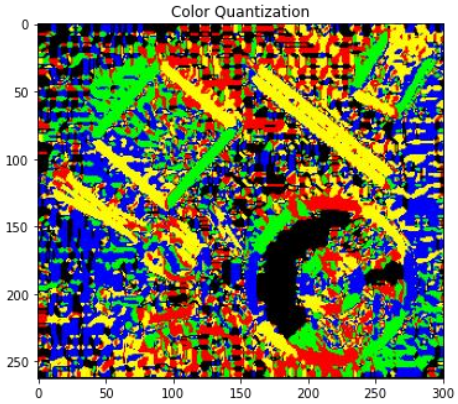


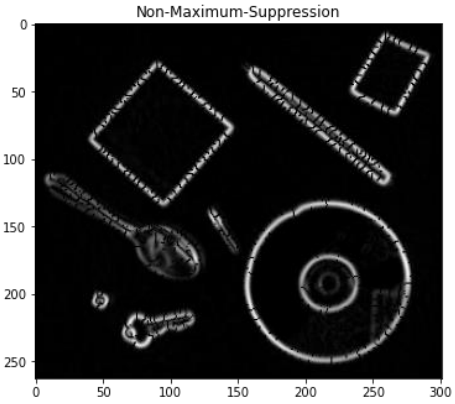
Image after Smoothing

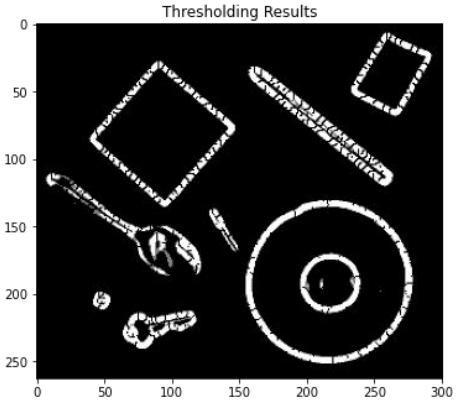

fx

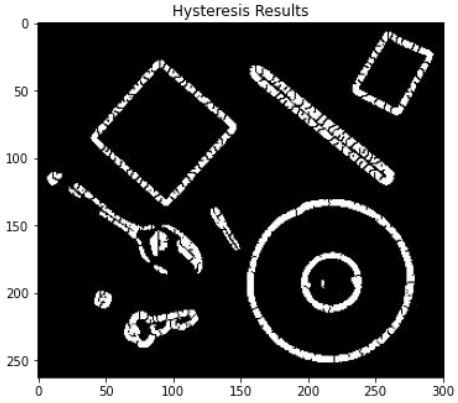

fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results


Hysteresis Results

Experiment 3: Sigma = 2 , T = 0.3 and filter (7,7)

LT=50 and HT=70

Image: Circle

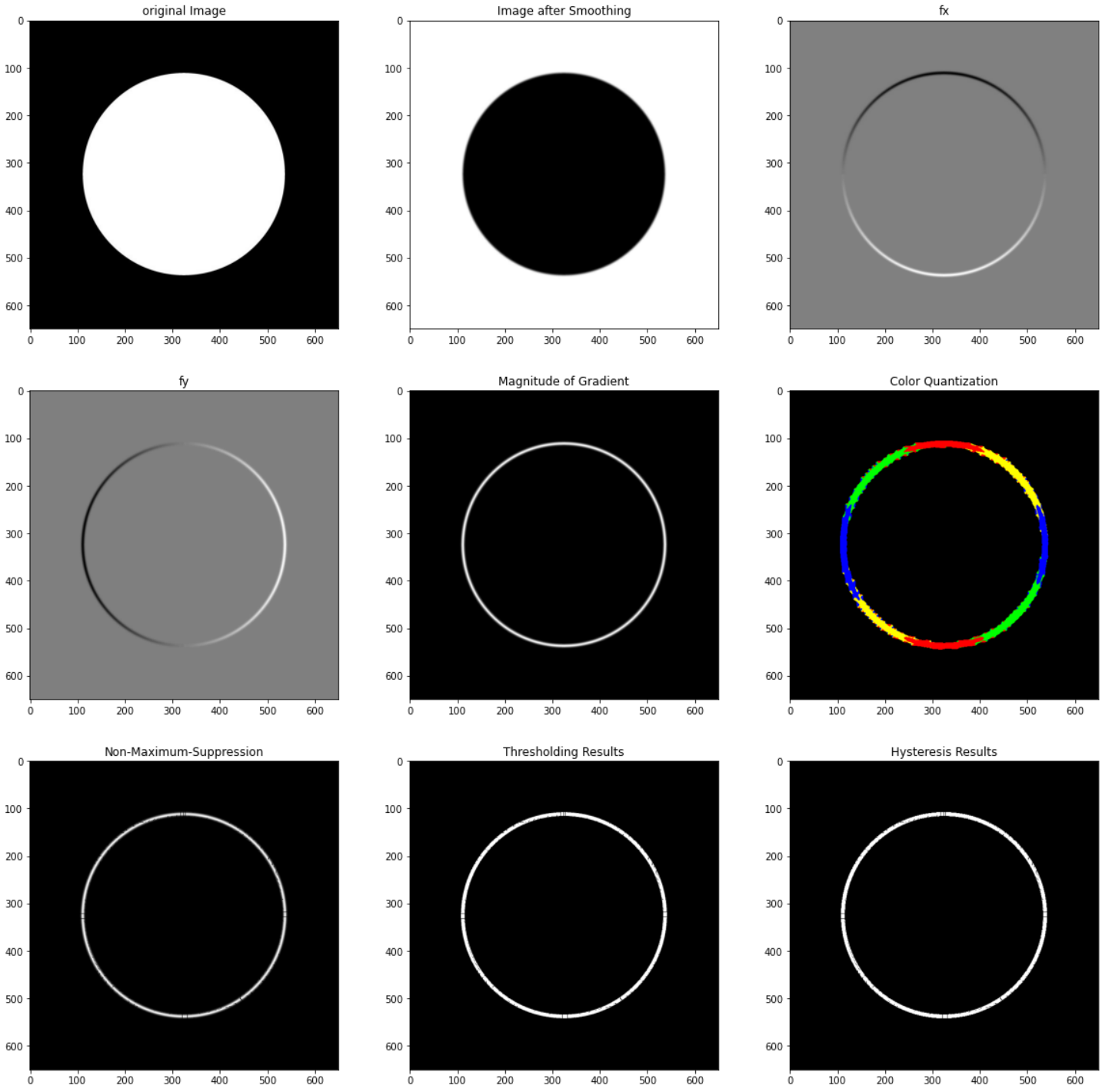
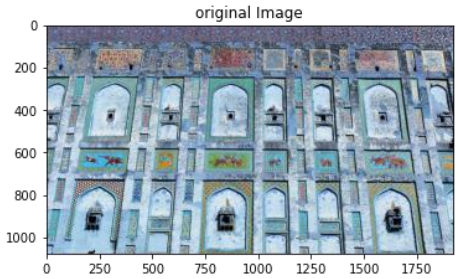
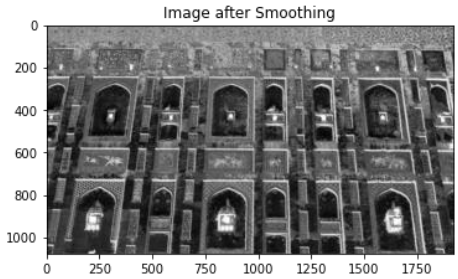
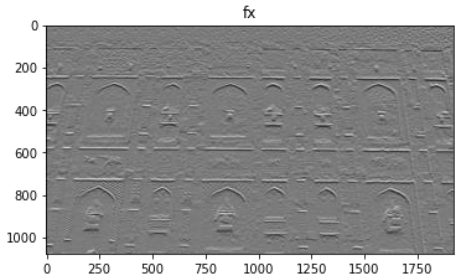
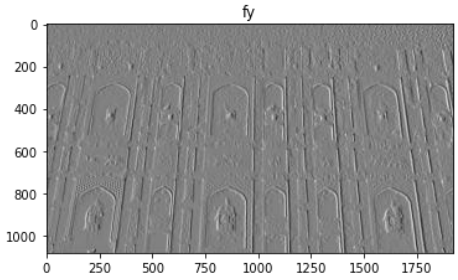


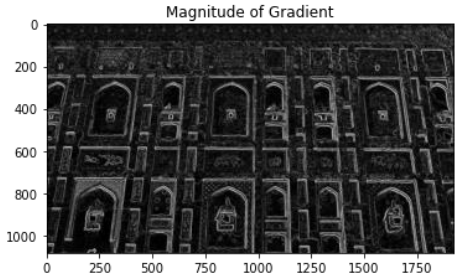
Image: wall Lahore

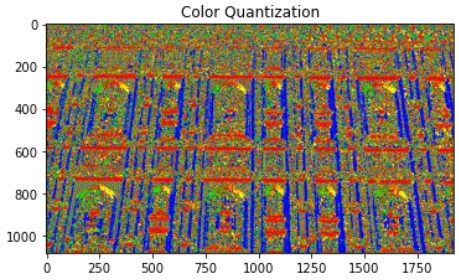


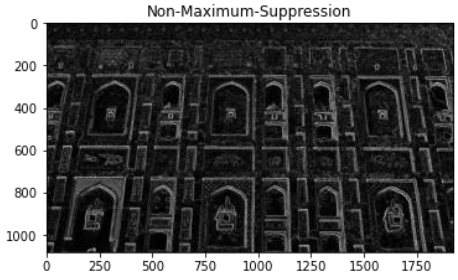


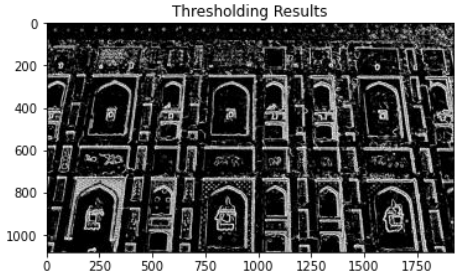












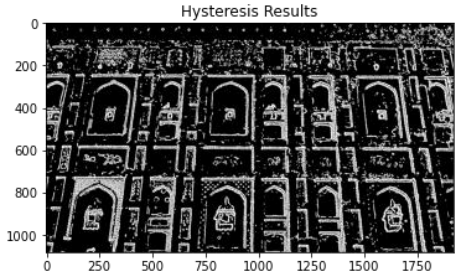
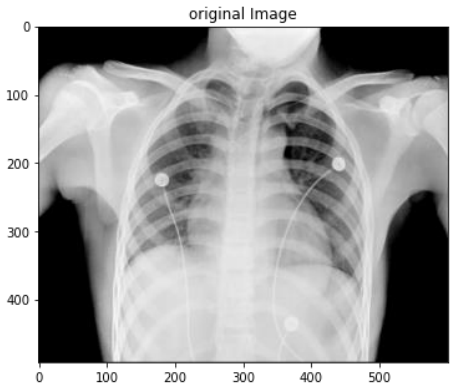


Image: Xray


original Image

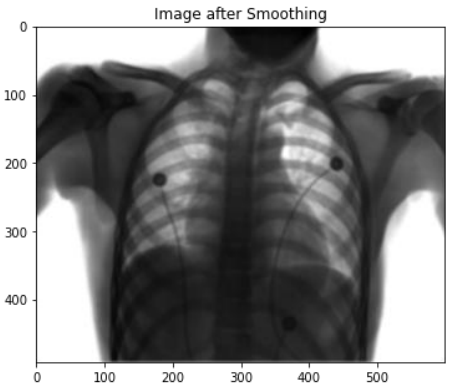
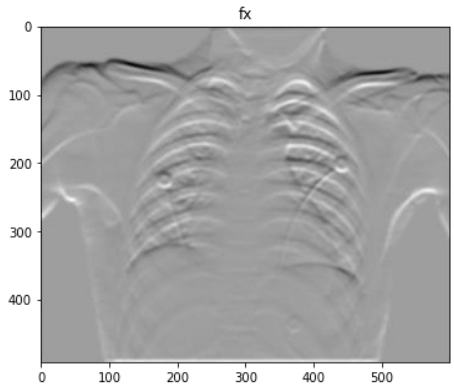
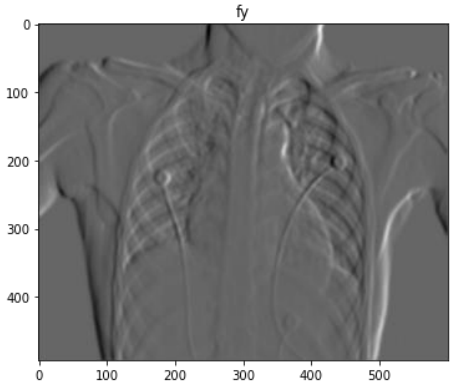
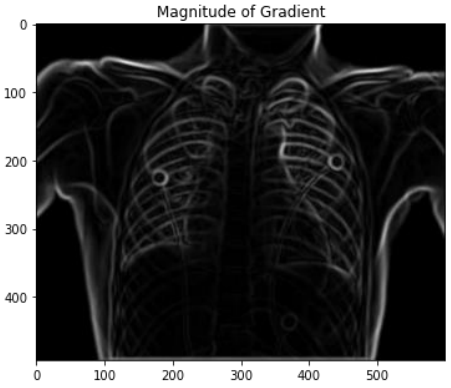
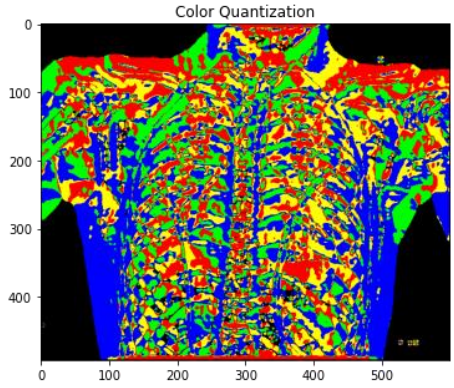


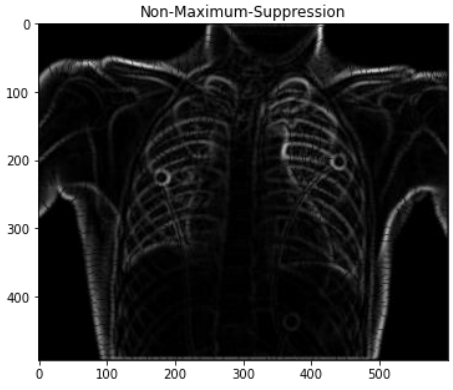
Image after Smoothing

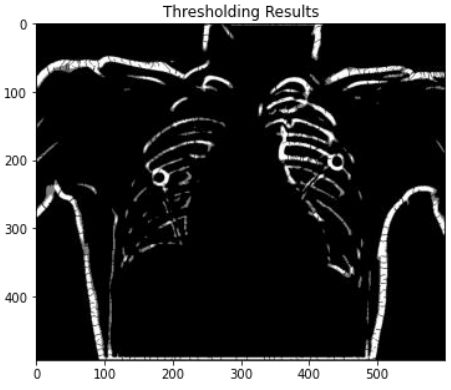

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

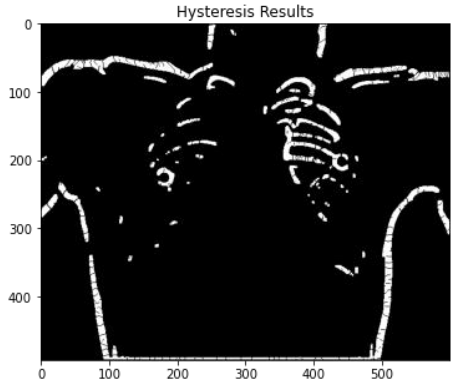

Hysteresis Results

Image: Ct scan

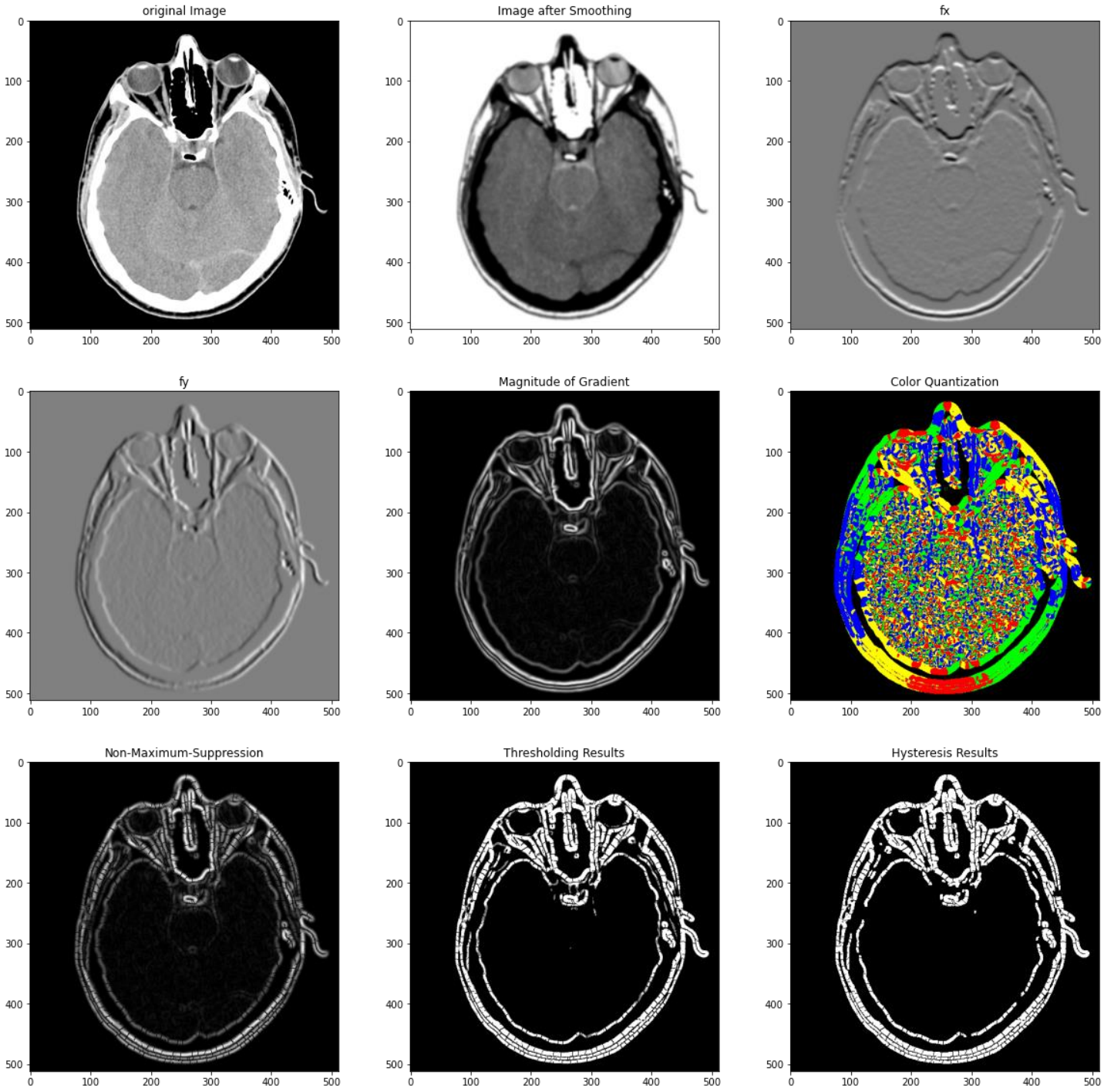
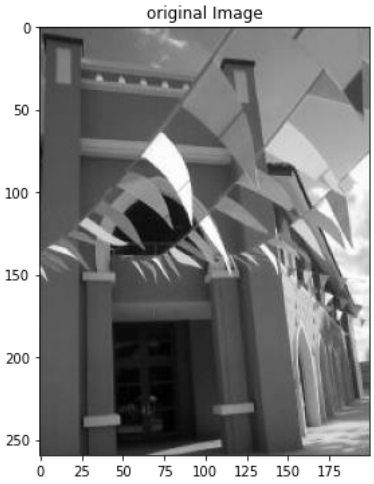


Image: mecca


original Image

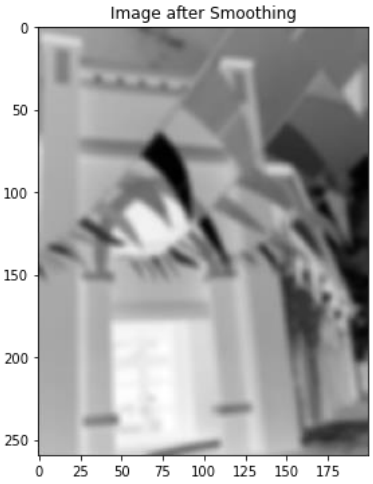
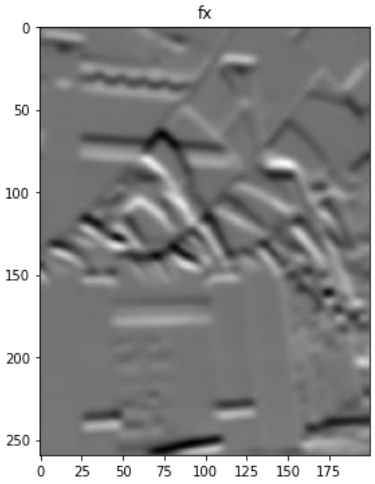
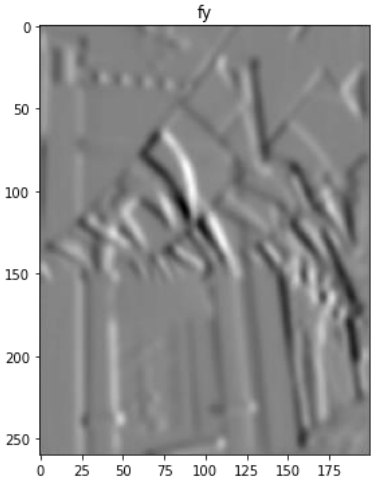
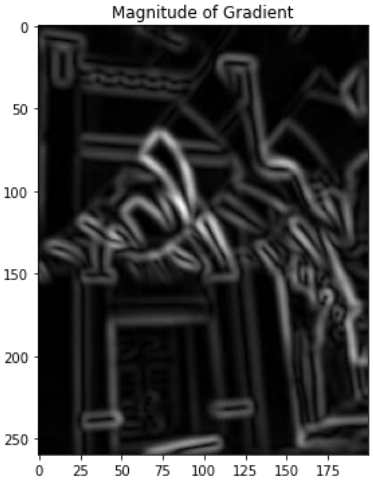
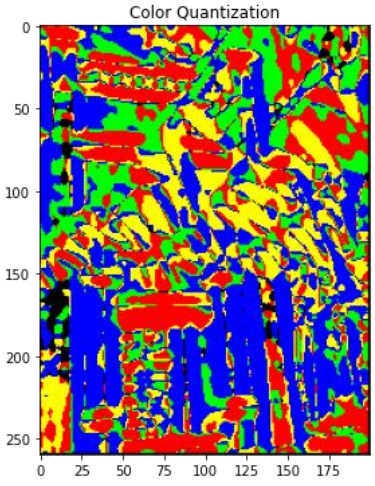


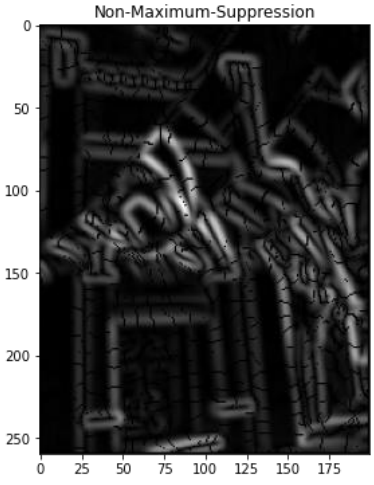
Image after Smoothing

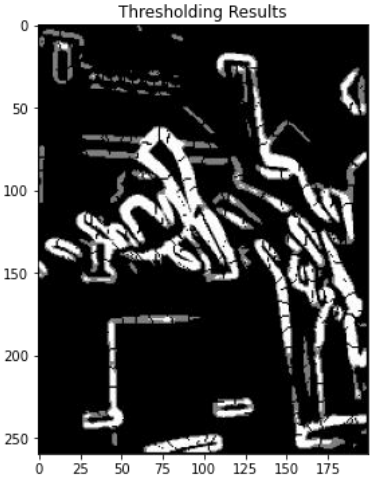

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

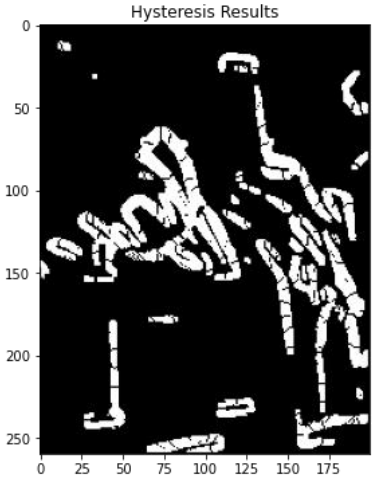
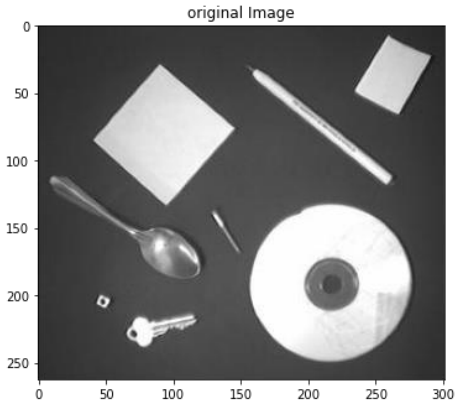
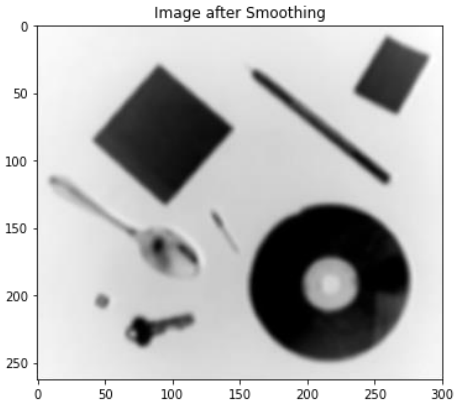
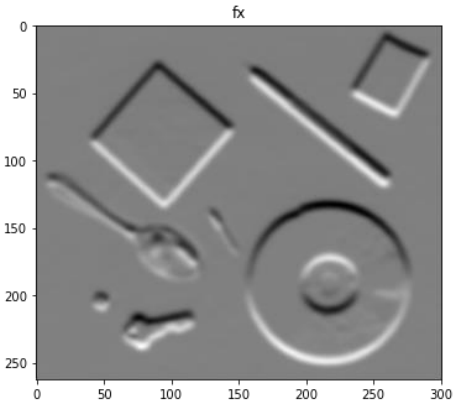
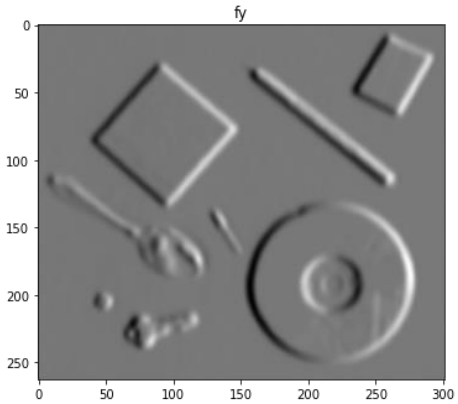

Hysteresis Results

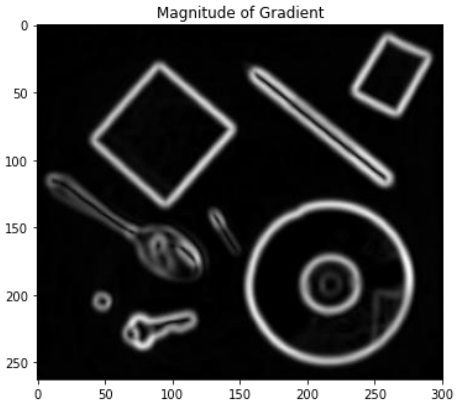
Image: Shape

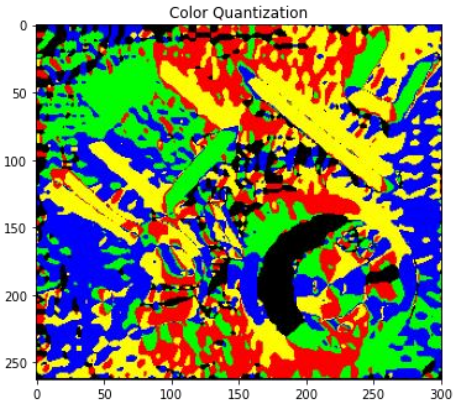


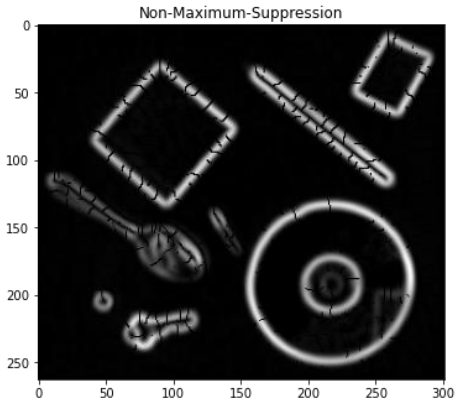


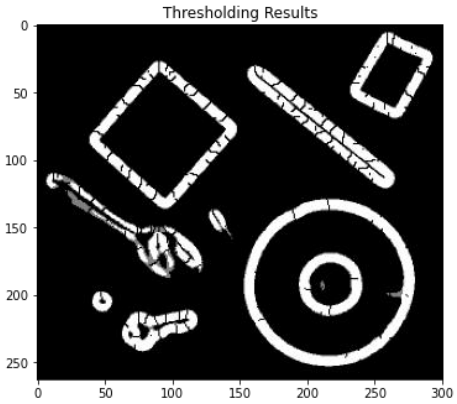


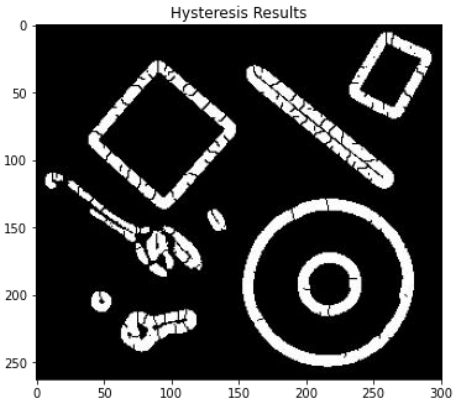












Discussion:

- Above all 3 experiments are performed on images with sigma (0,5,1,2) with HT and LT pair(70,50).
- Vertical and Horizontal edges can be seen by fy and fx respectively.
- Non-Maximum-Suppression is making our edges thin which can be seen from the plot of Non-Maximum-Suppression results.
- The thresholding plot shows both weak edges as gray and strong edges as white.
- Hysteresis Thresholding transforming all weak edges into strong ones according to the specified criteria
- One different effect of the small and larger filter can be seen in images that the edges of the images with large seem to be more smearing out of the pixel value than the 3x3 filter.

Experiment 4: Sigma = 0.5, T =0.3 and filter= (3, 3)

LT=30 and HT= 100

Image: Circle

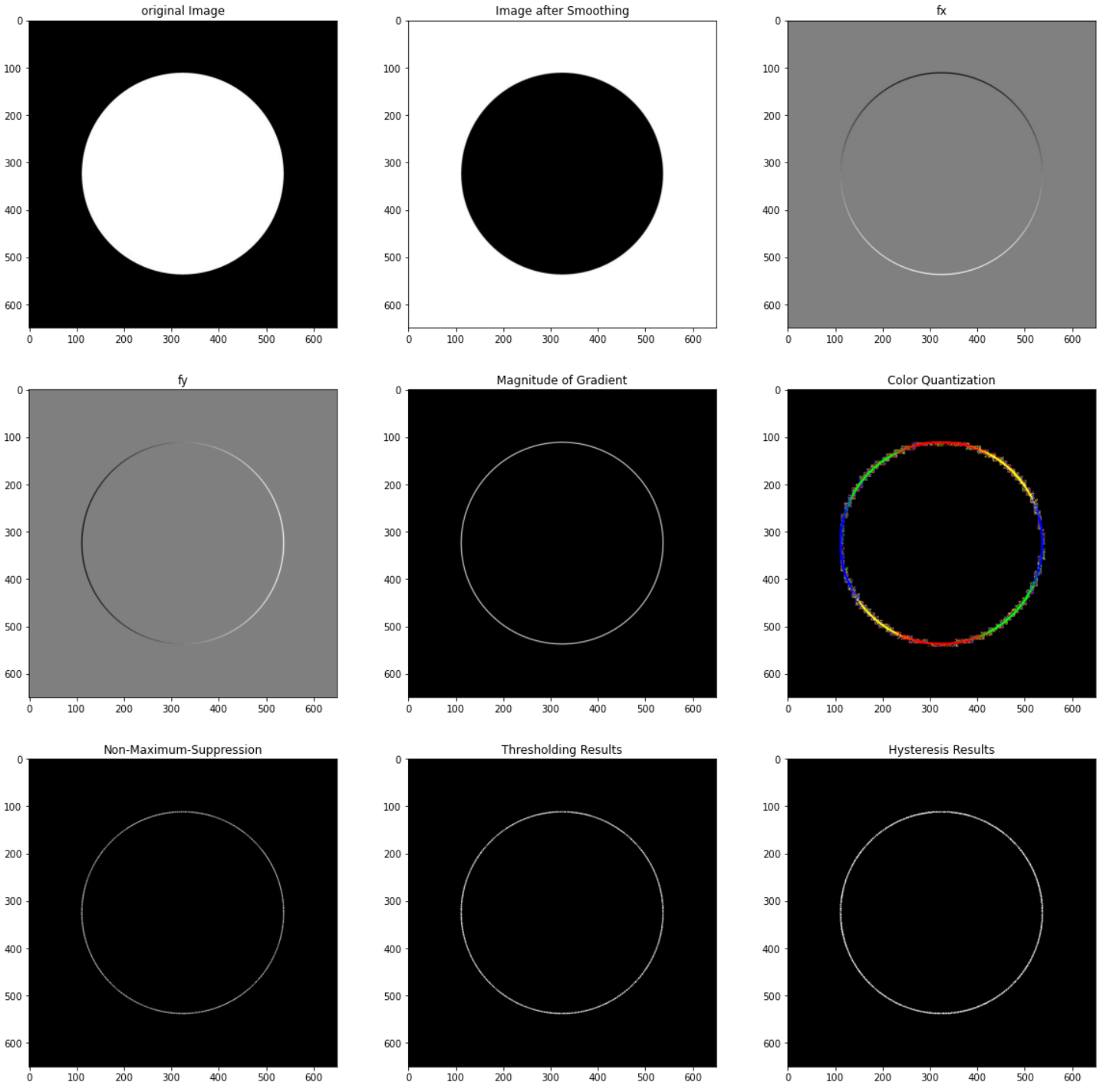
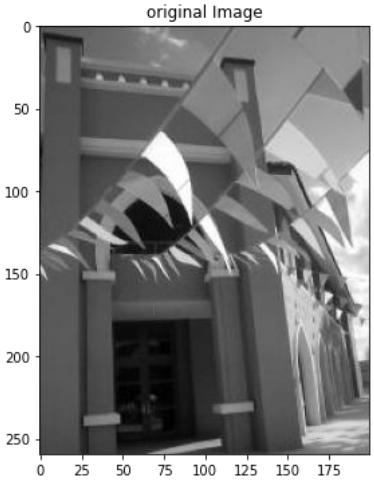


Image: mecca


original Image

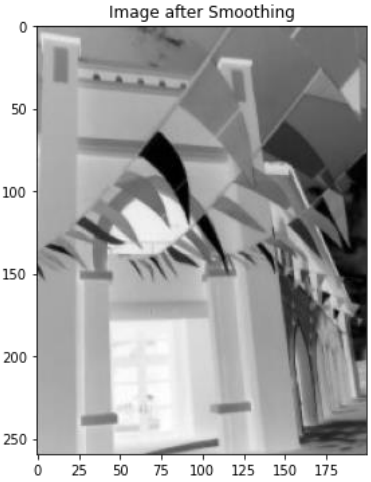
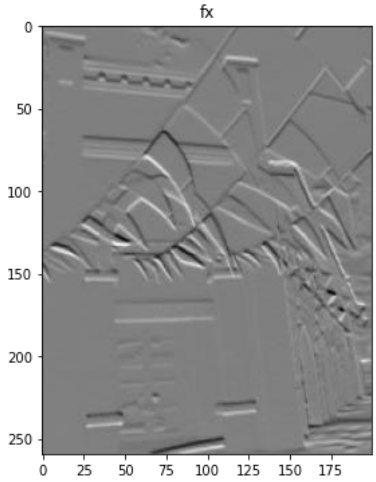
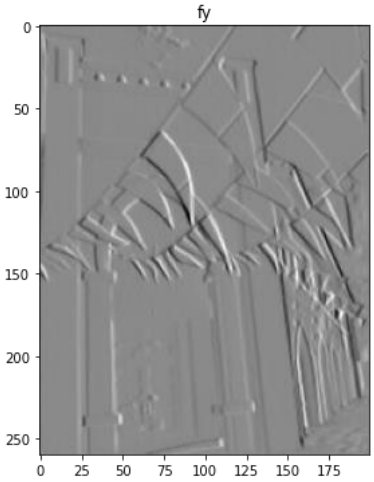
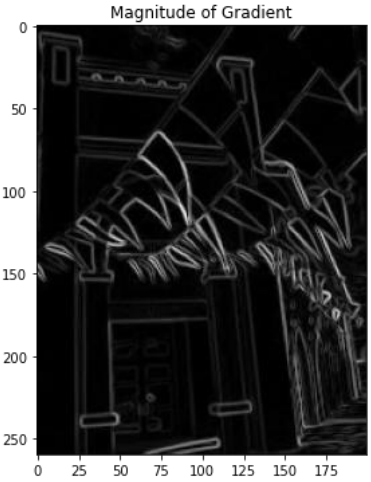
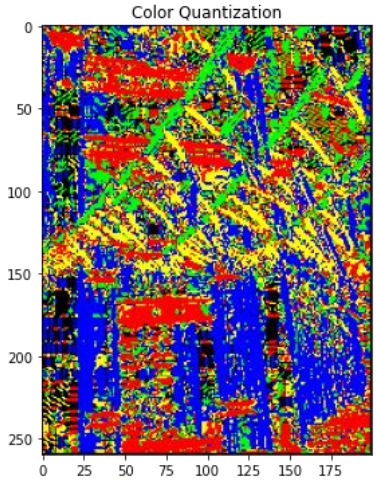


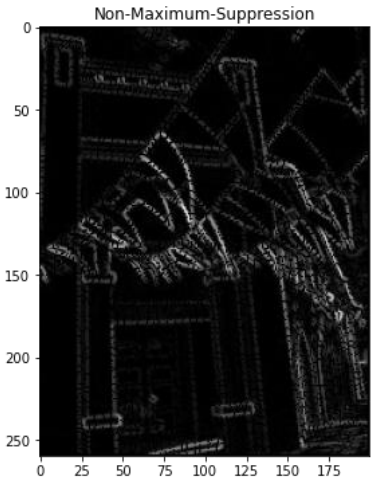
Image after Smoothing

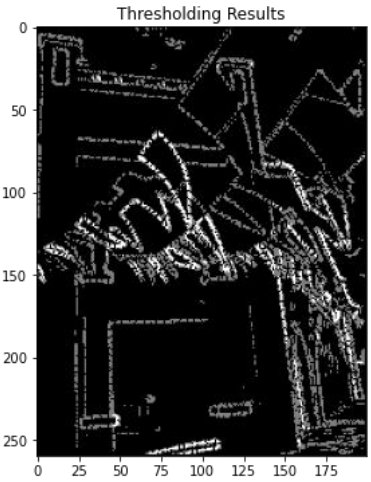

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

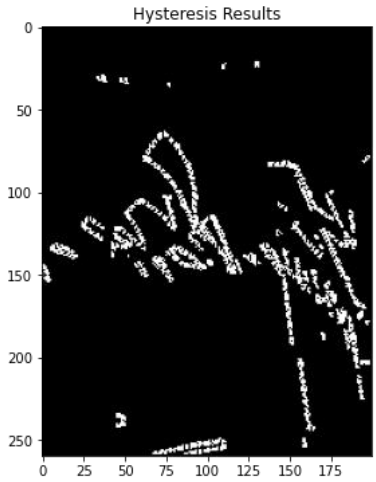
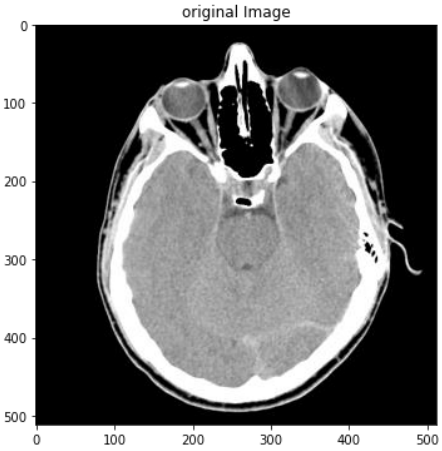
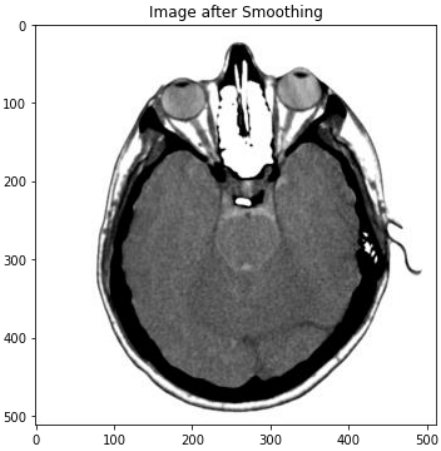
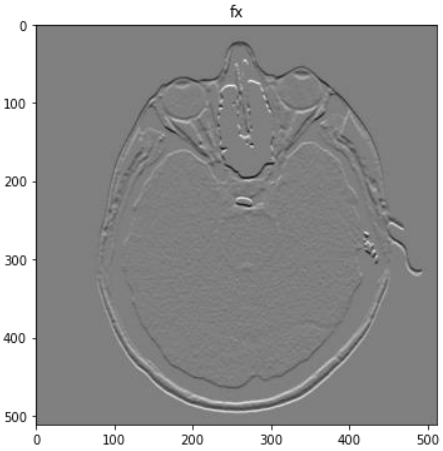
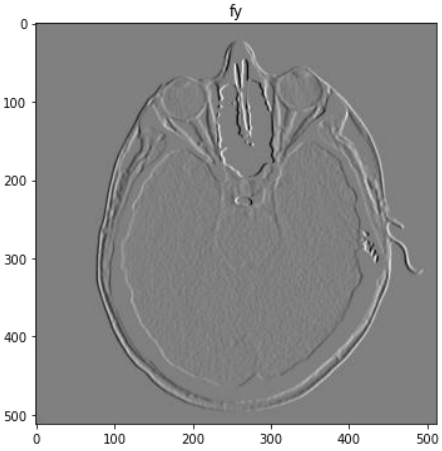

Hysteresis Results

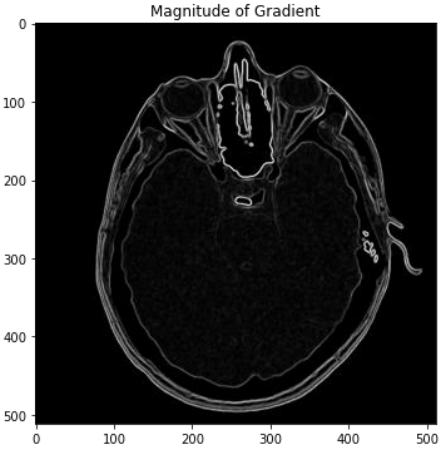
Image: CT scan

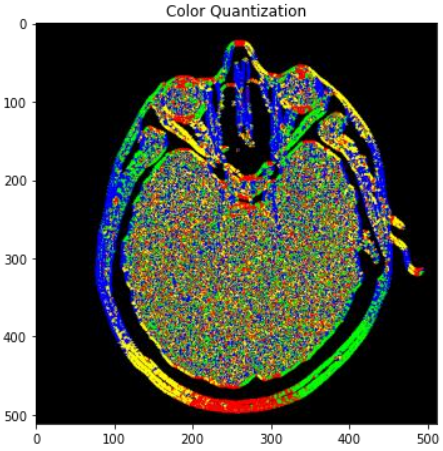


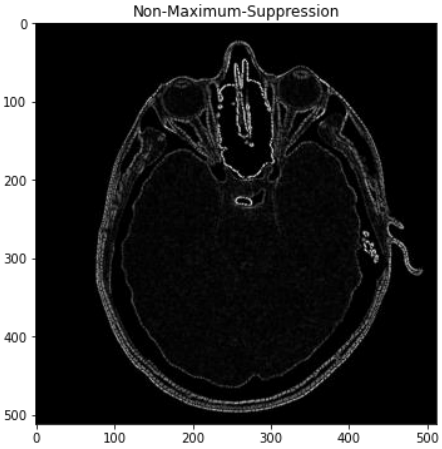


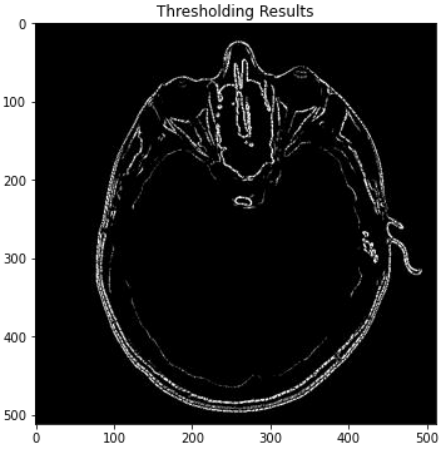












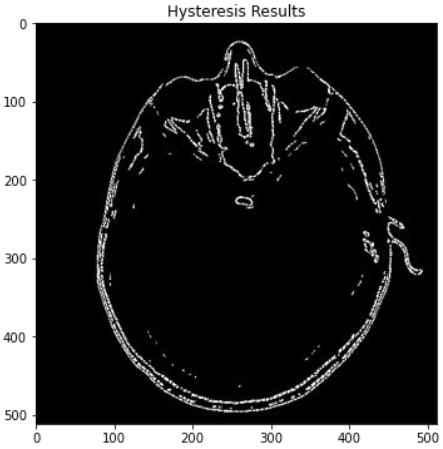


Image: x-ray

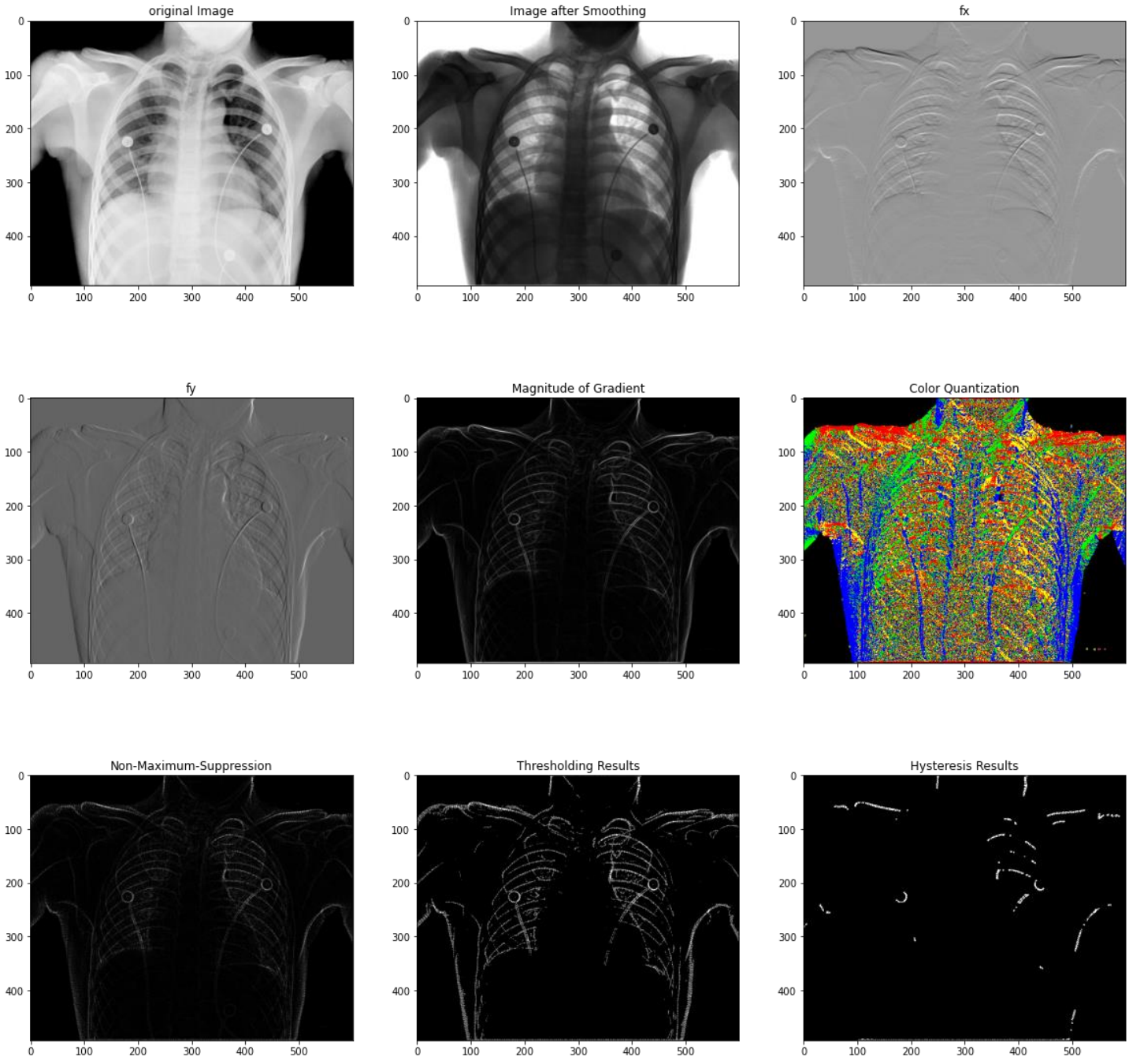
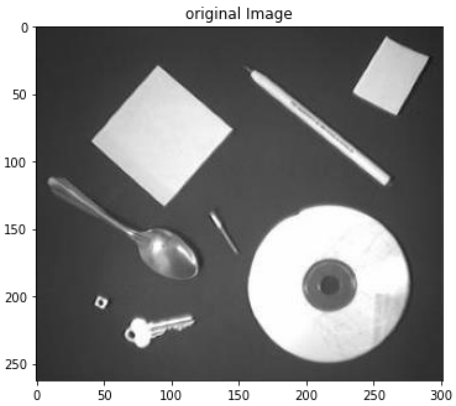

original Image
Image after Smoothing
fx
fy
Magnitude of Gradient
Color Quantization
Non-Maximum-Suppression
Thresholding Results
Hysteresis Results

Image: Shape


original Image

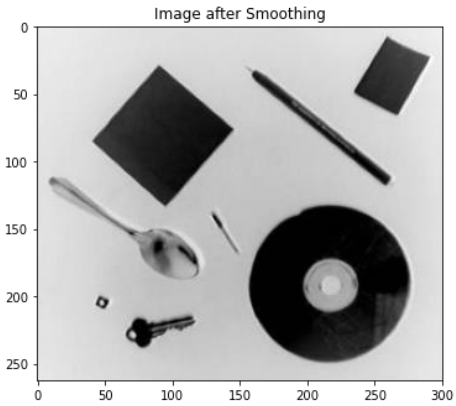
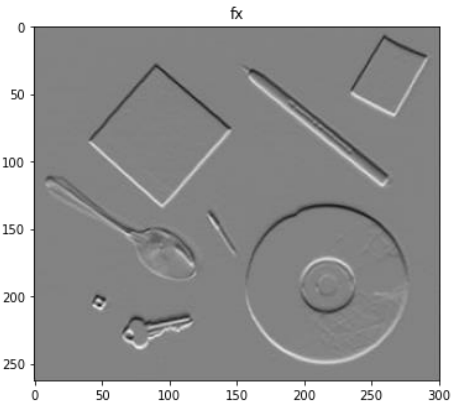
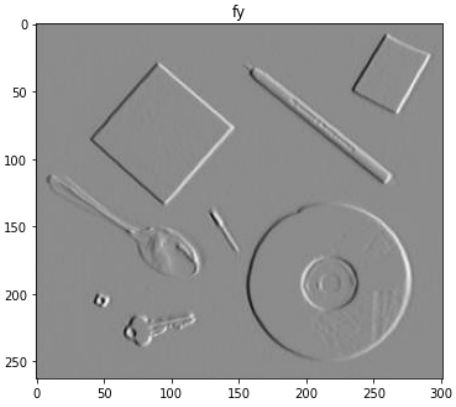
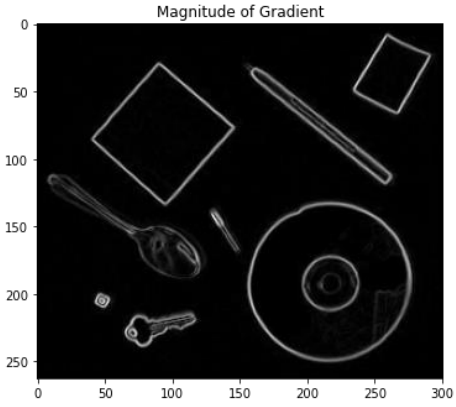
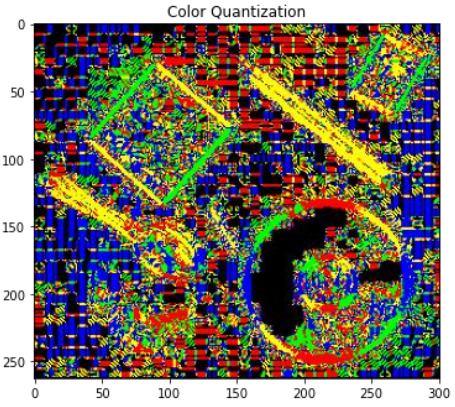


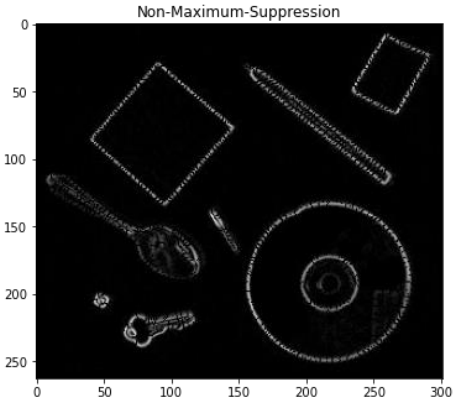
Image after Smoothing

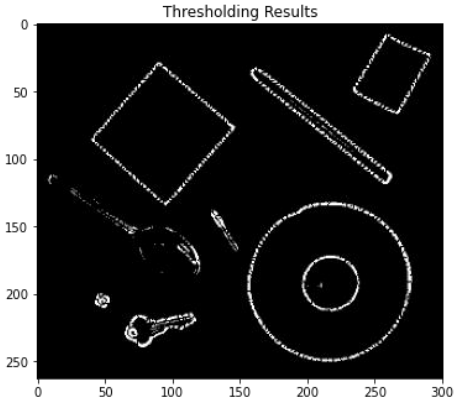

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

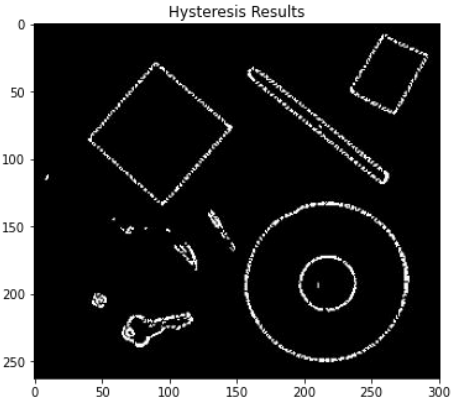
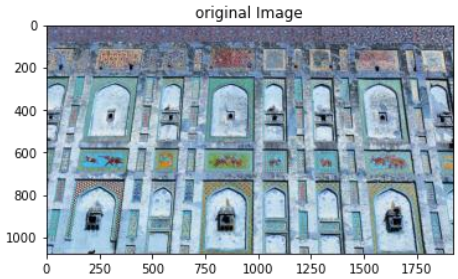
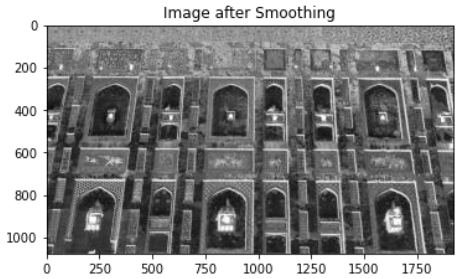
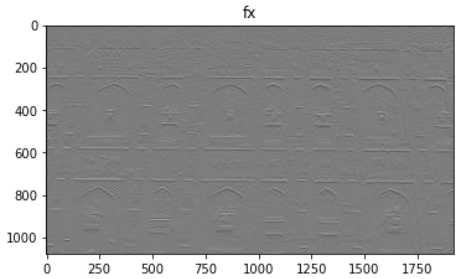
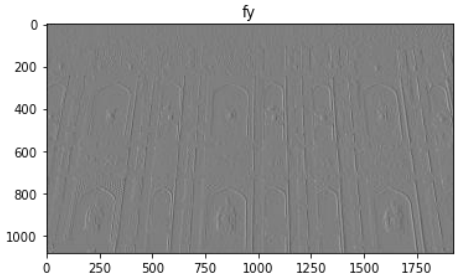

Hysteresis Results

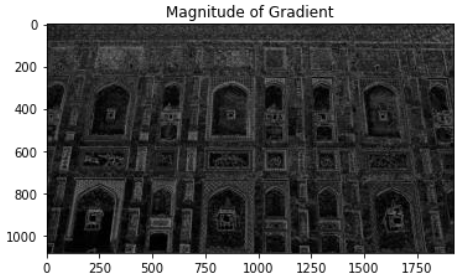
Image: wall Lahore

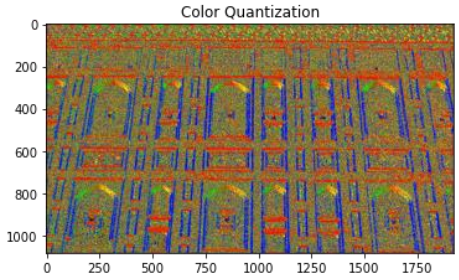


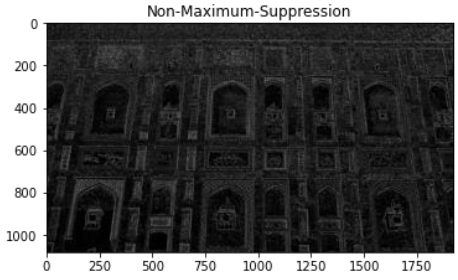


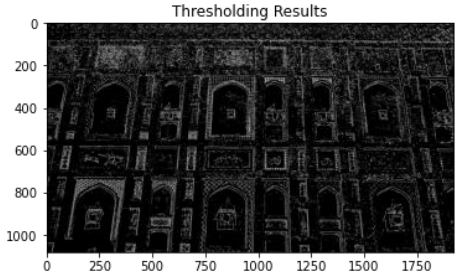


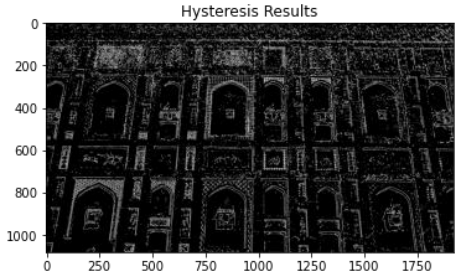












Experiment 4: Sigma = 1, T =0.3 and filter= (5, 5)

LT=30 and HT= 100

Image: Circle

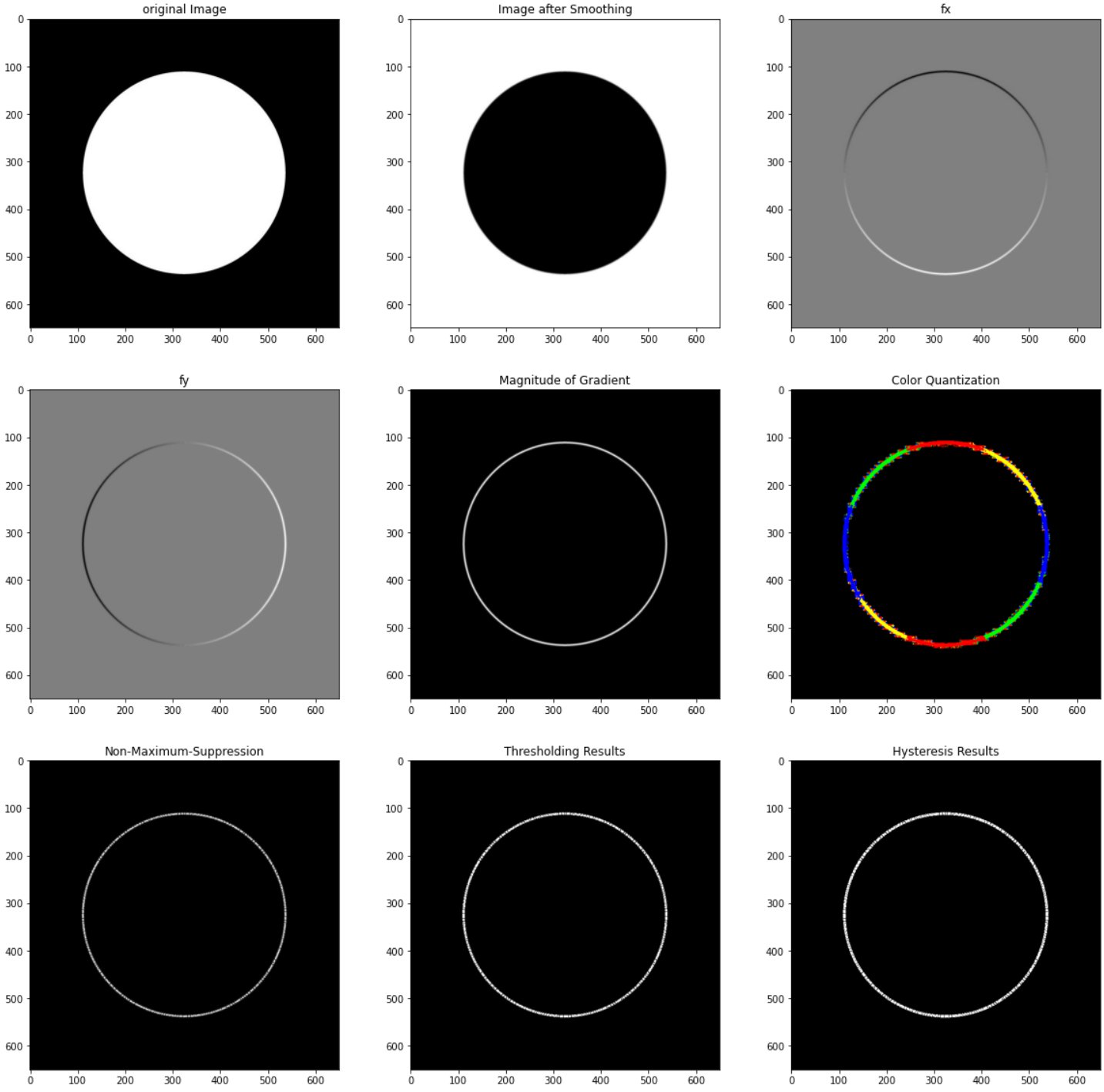
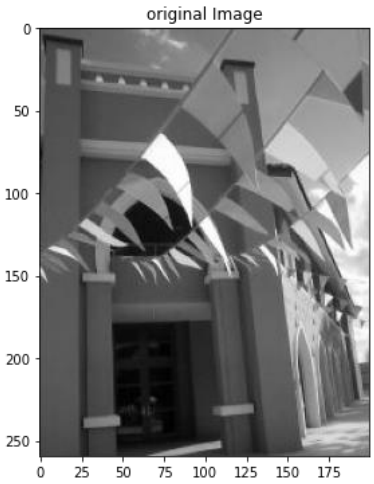


Image: mecca


original Image

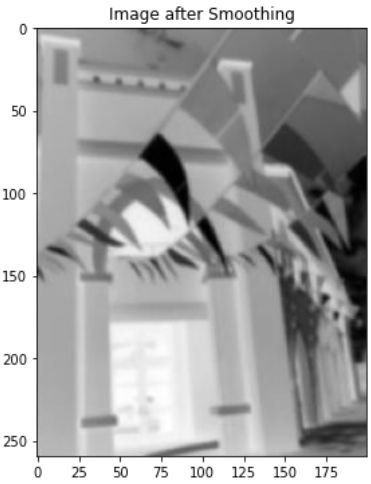
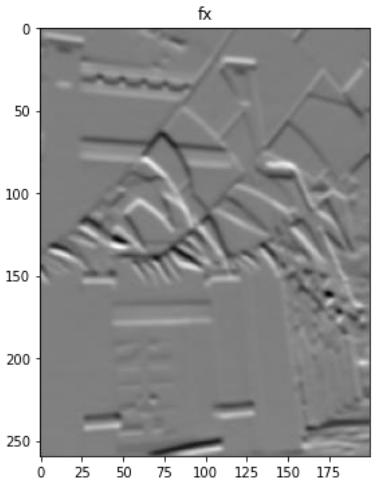
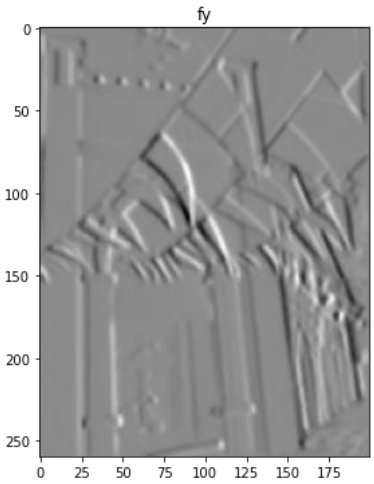
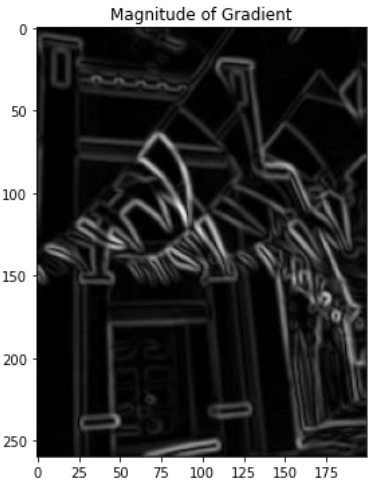
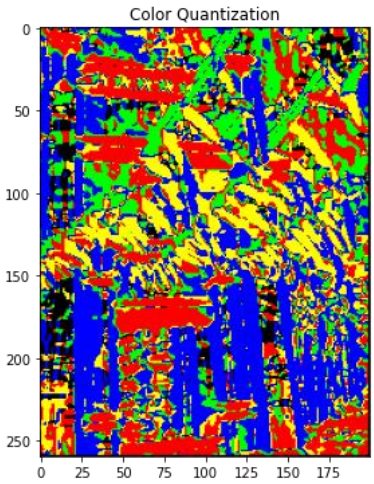


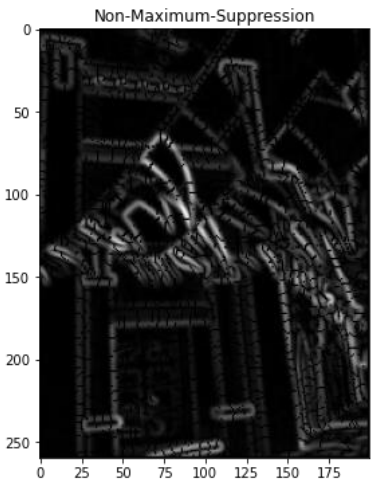
Image after Smoothing

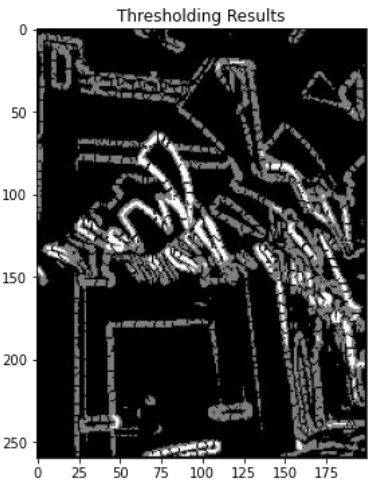

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

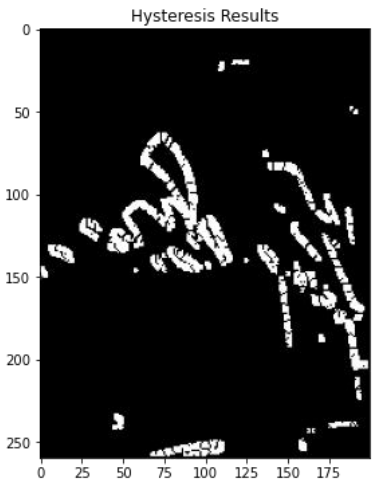
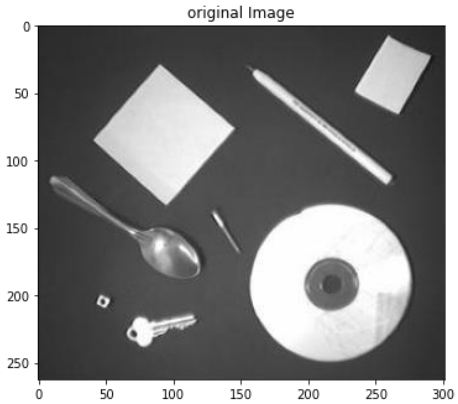

Hysteresis Results

Image: Shape


original Image

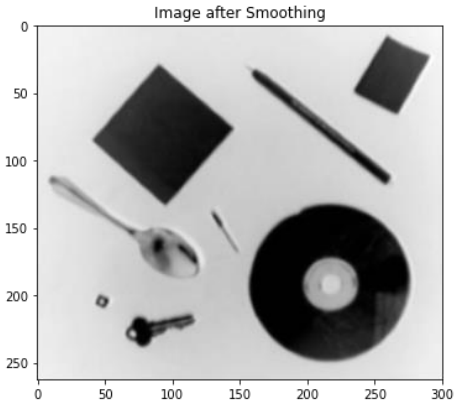
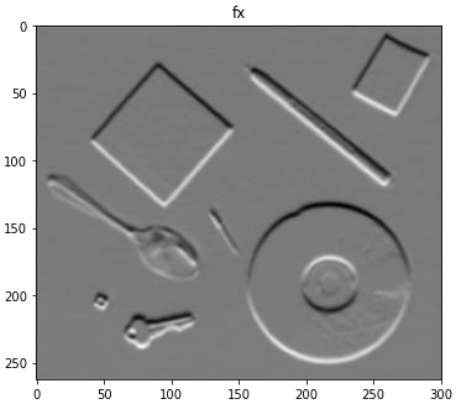
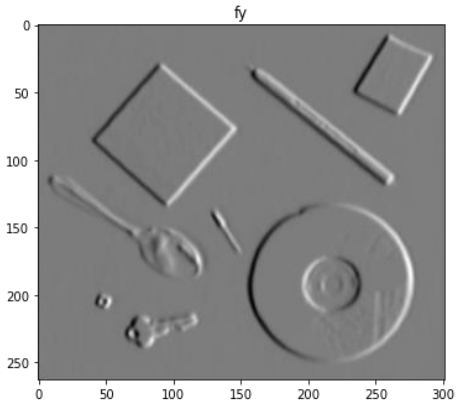
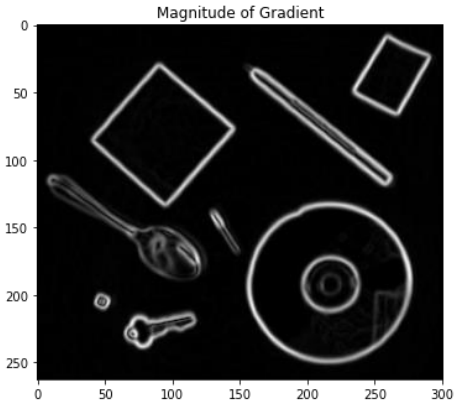
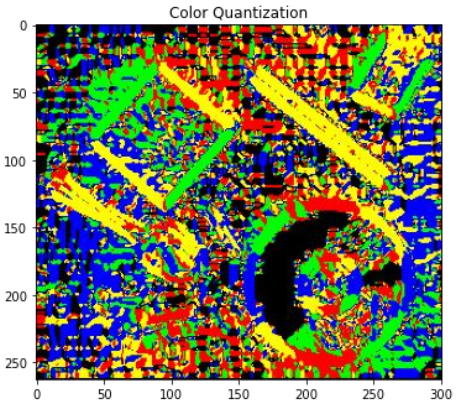


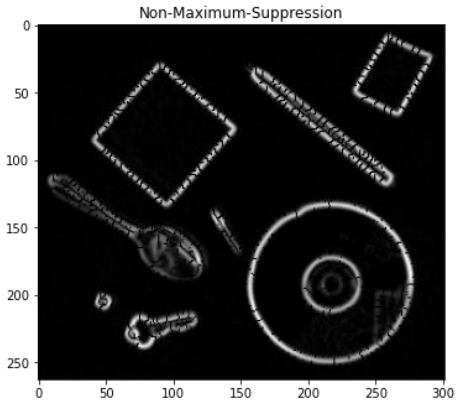
Image after Smoothing

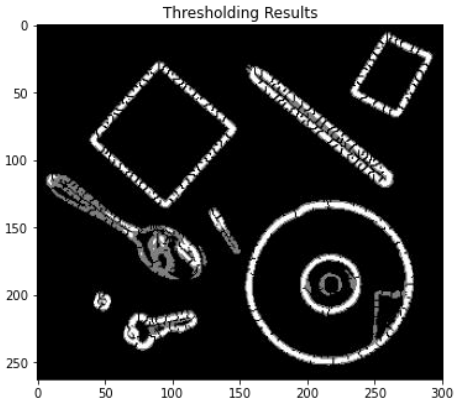

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

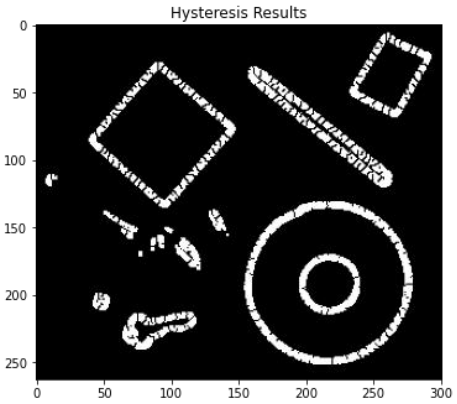

Hysteresis Results

Image: x-ray

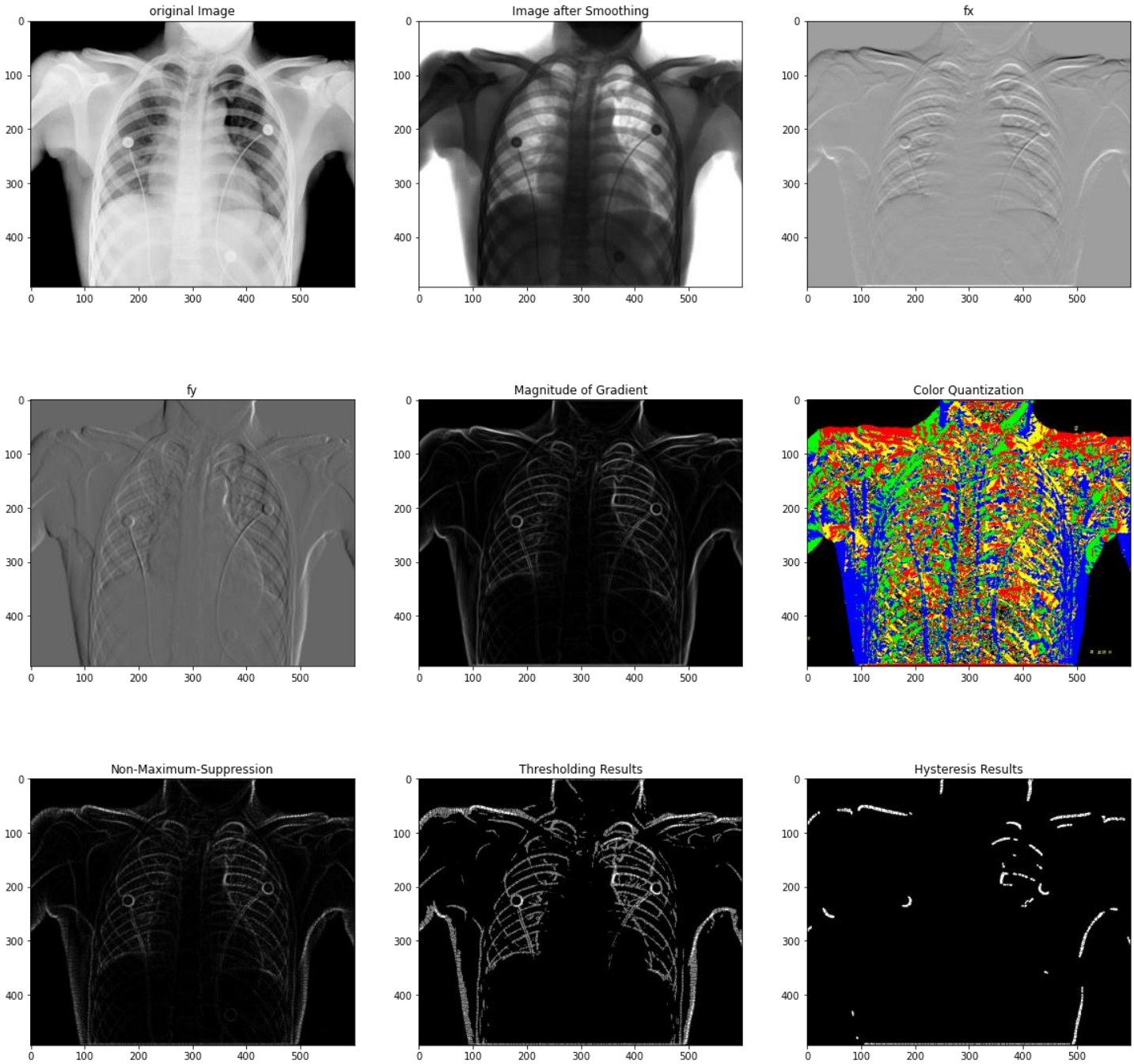

original Image
Image after Smoothing
fx
fy
Magnitude of Gradient
Color Quantization
Non-Maximum-Suppression
Thresholding Results
Hysteresis Results

Image: CT scan

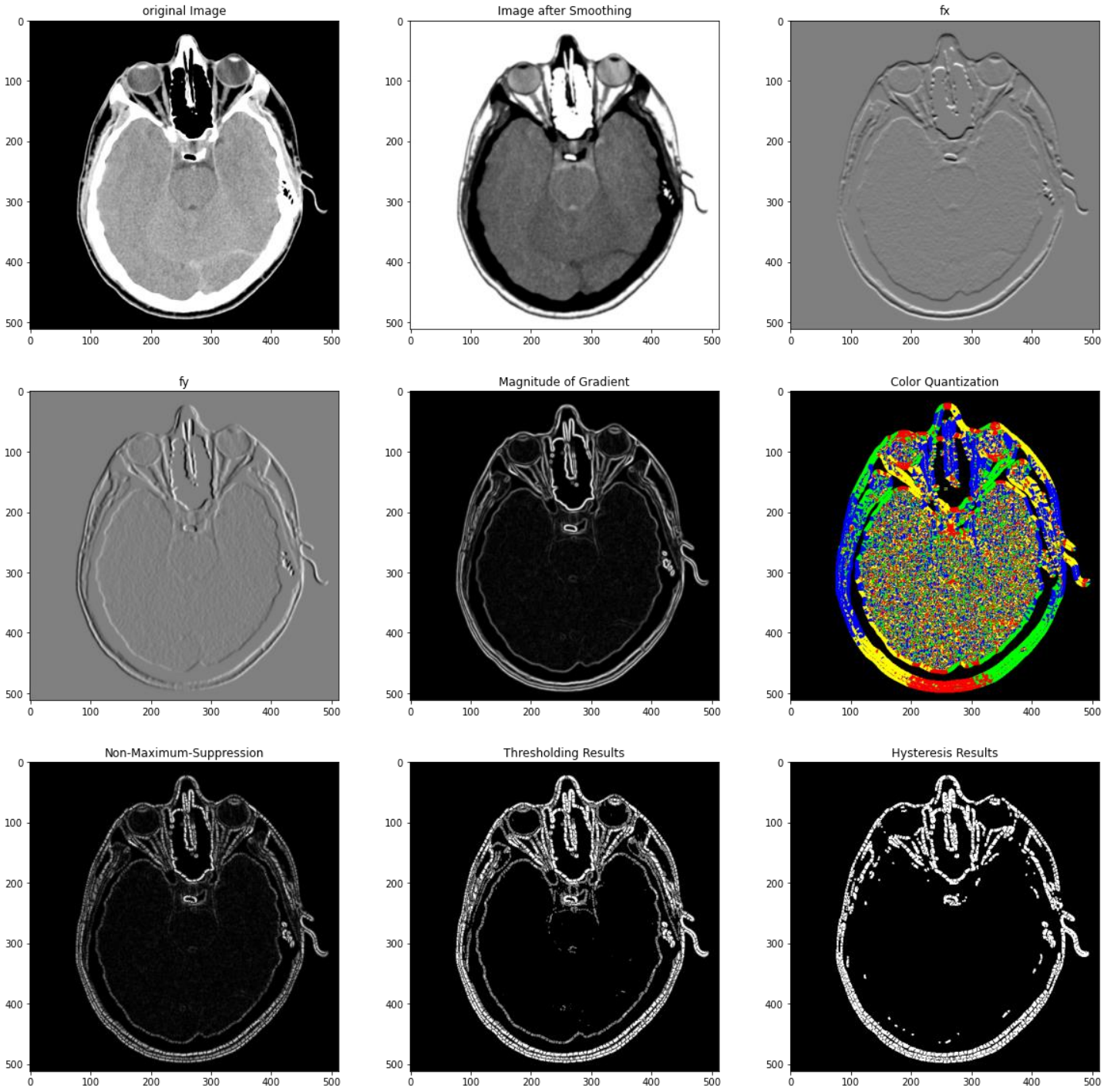
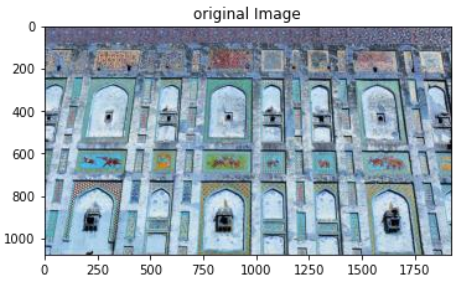
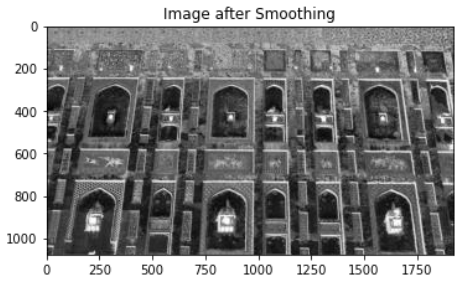
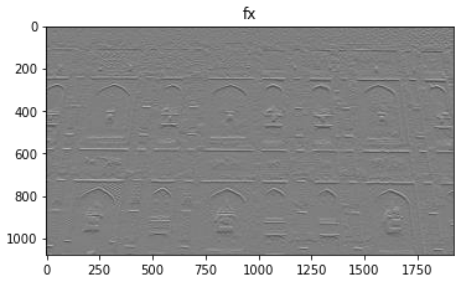
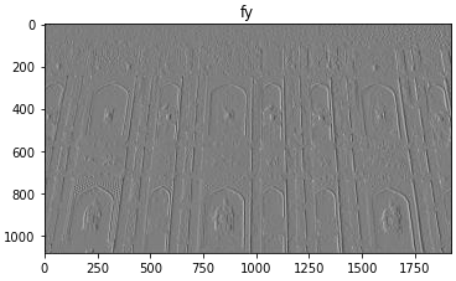


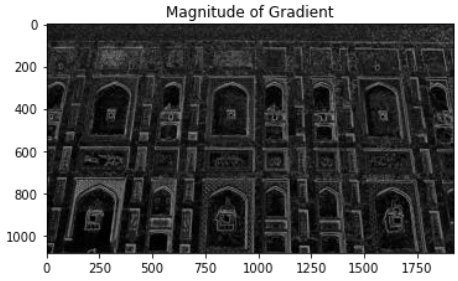
Image: wall Lahore

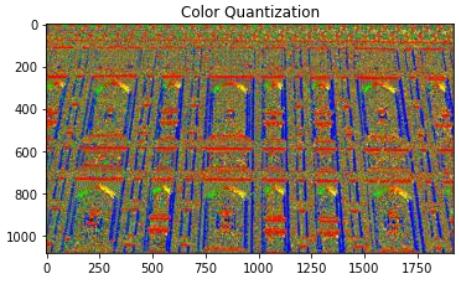


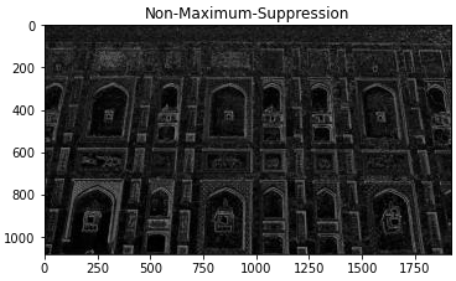


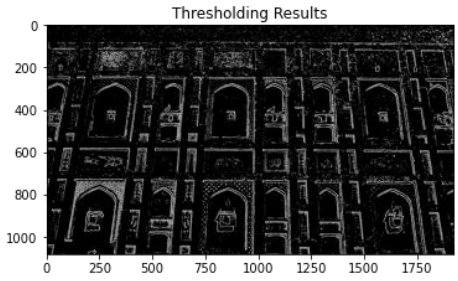


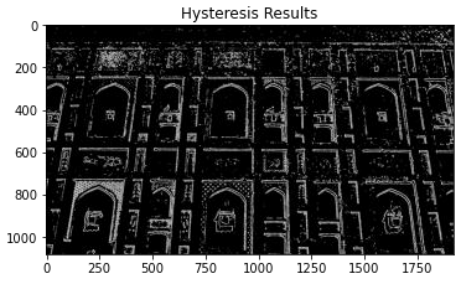












Experiment 6: Sigma = 2, T =0.3 and filter= (7, 7)

LT=30 and HT= 100

Image: Circle

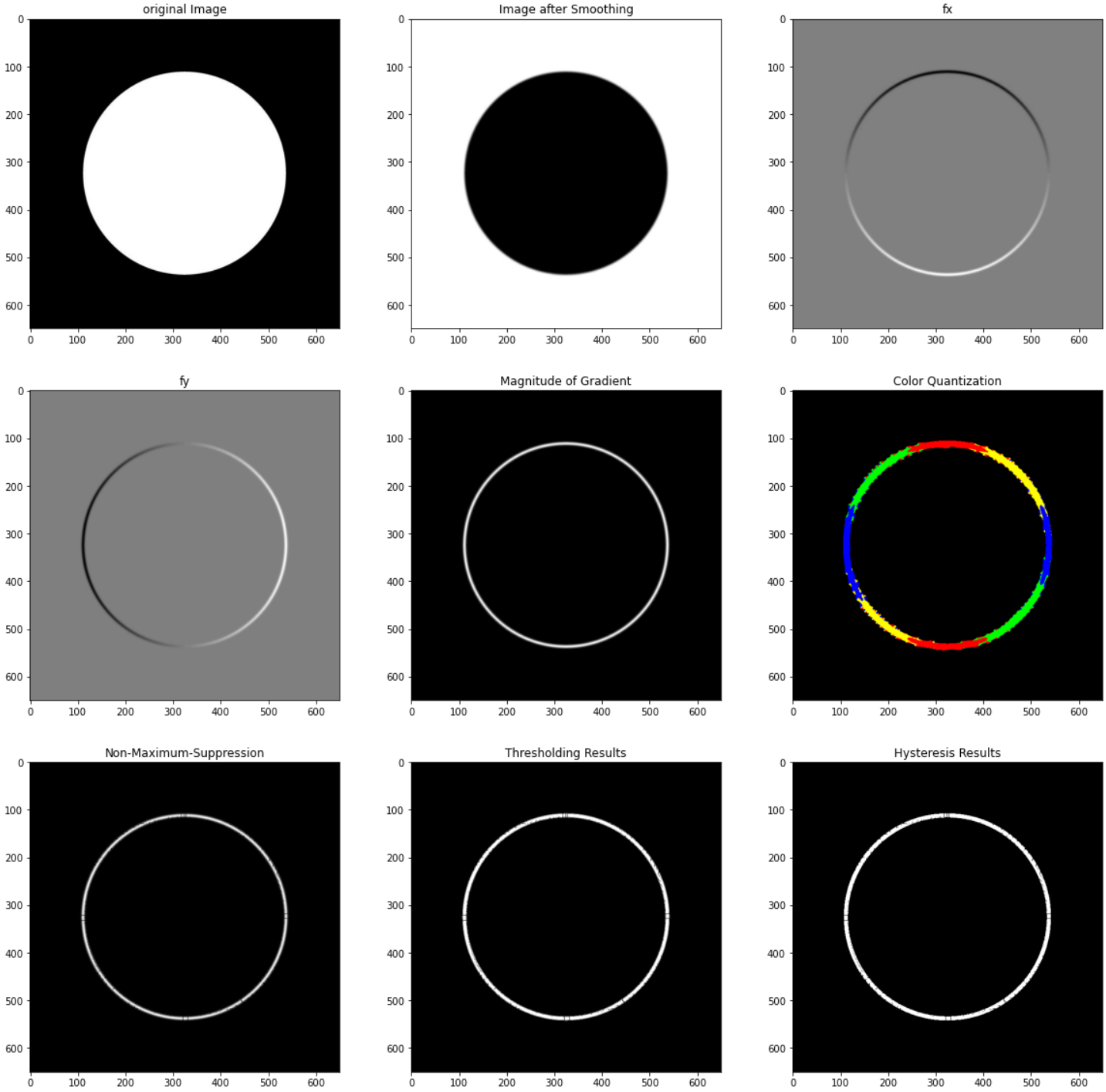
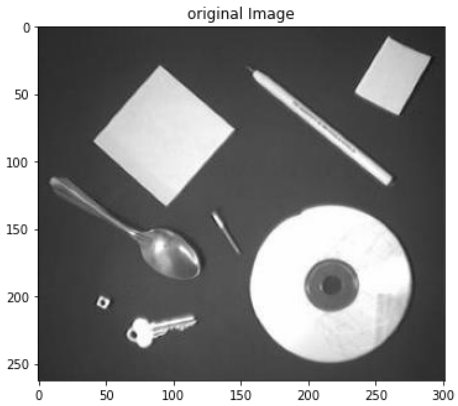


Image: Shape


original Image

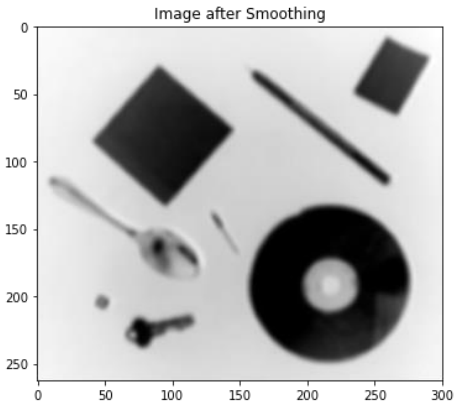
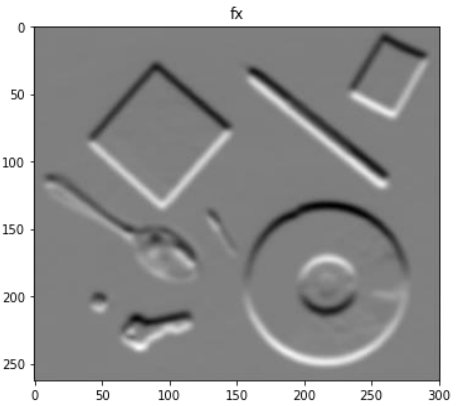
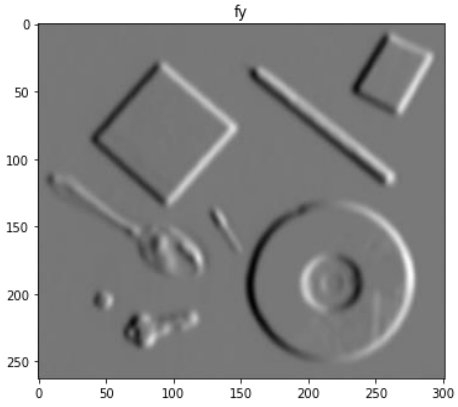
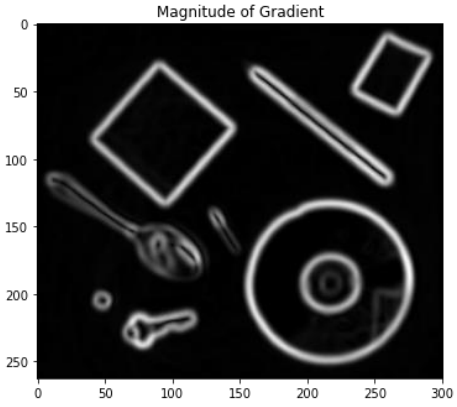
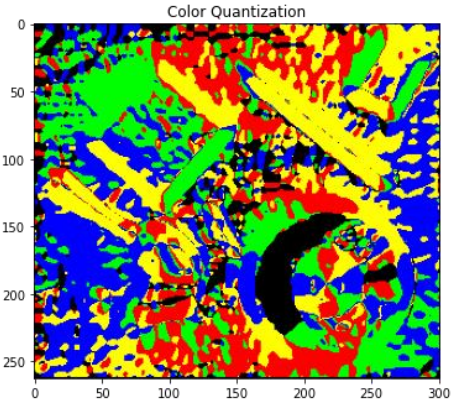


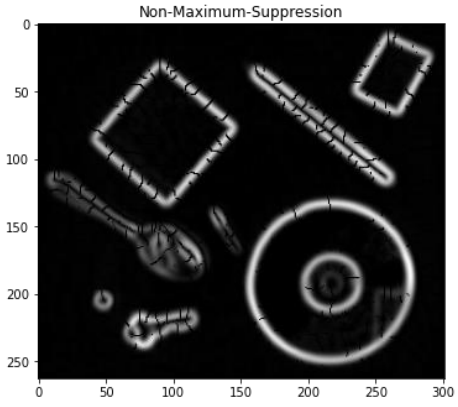
Image after Smoothing

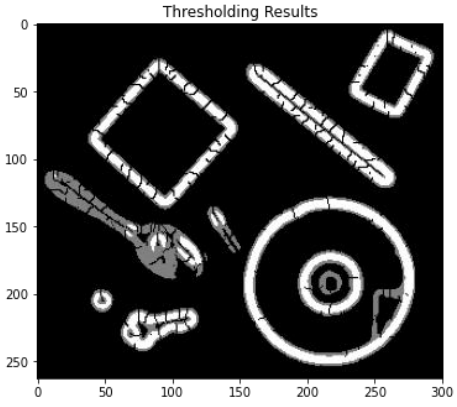

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

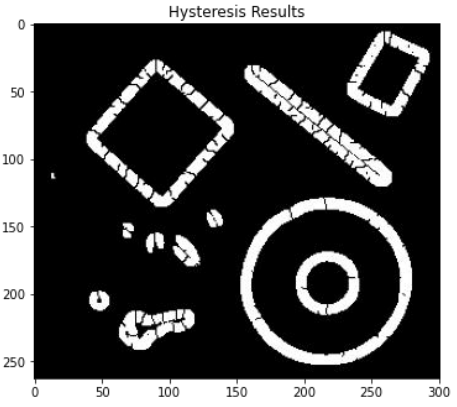

Hysteresis Results

Image: CT scan

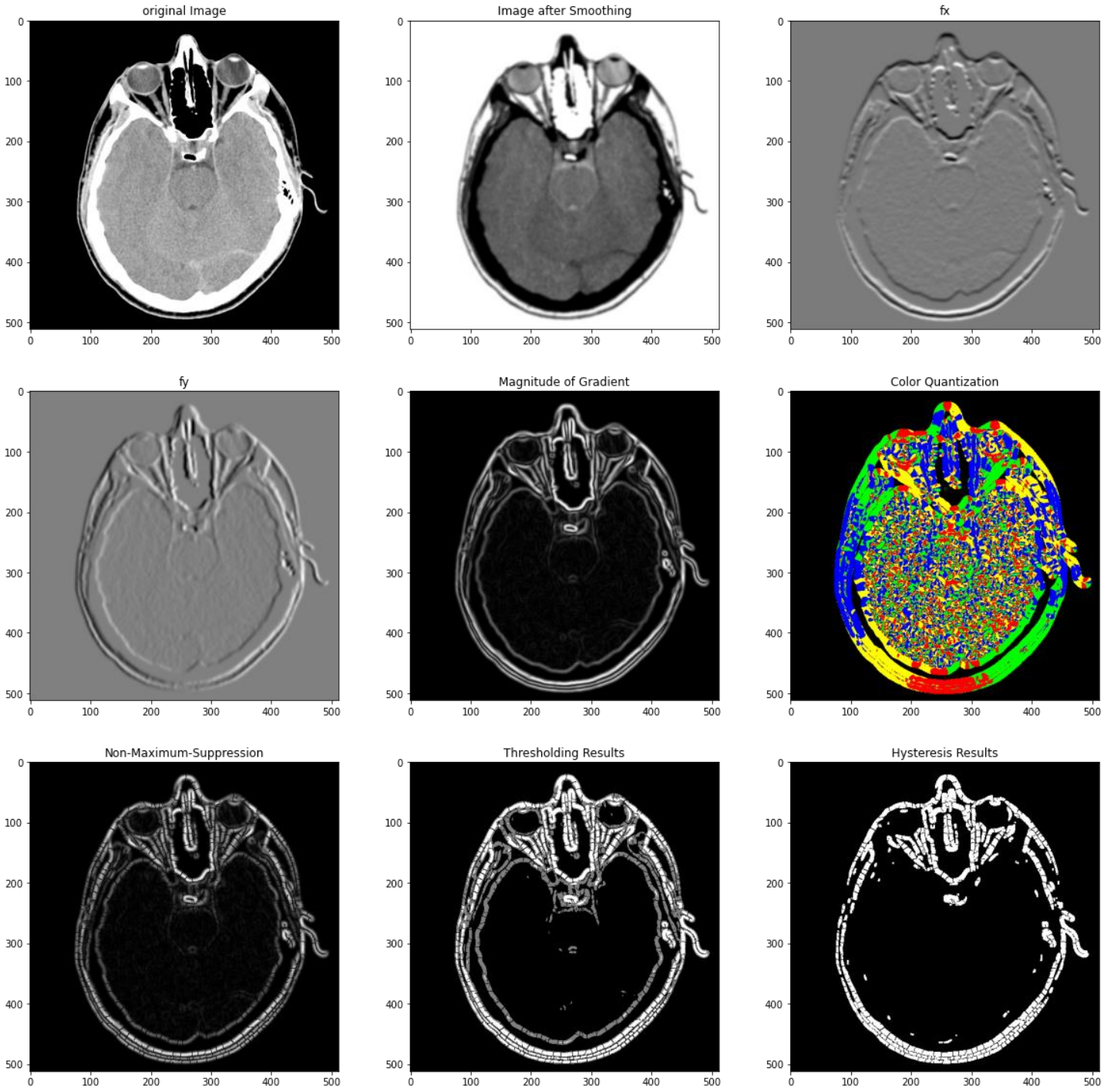
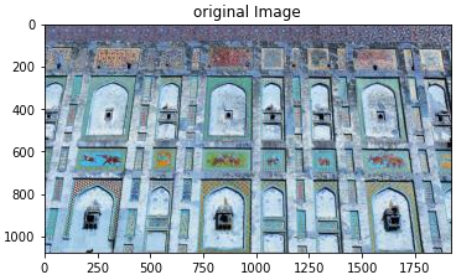
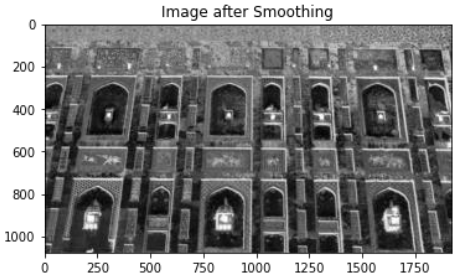
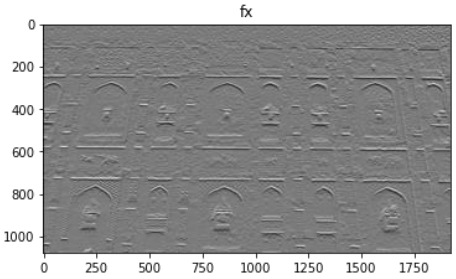
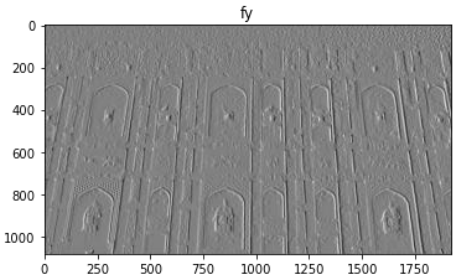


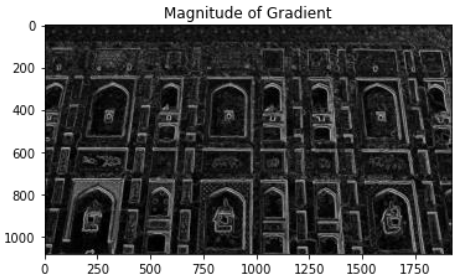
Image: wall Lahore

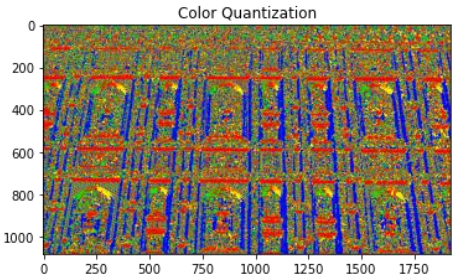


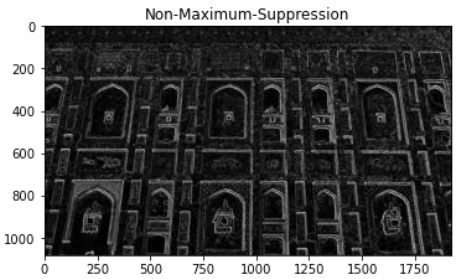


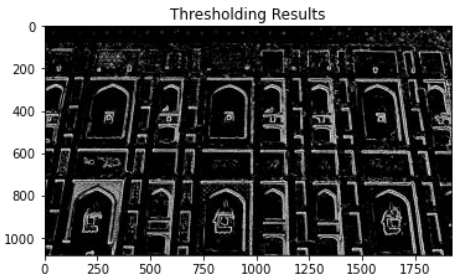












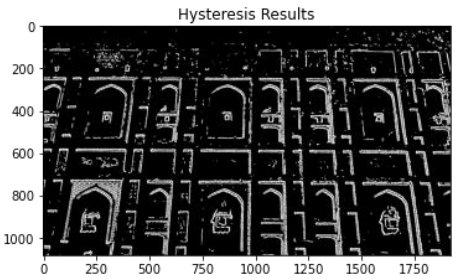
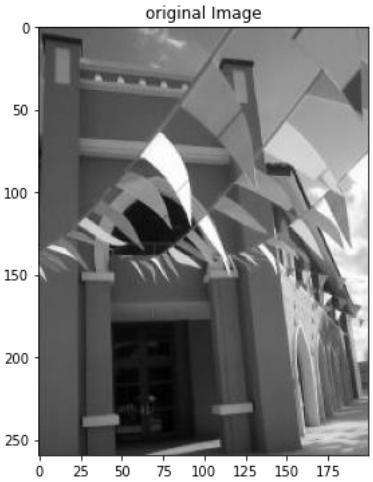


Image: mecca


original Image

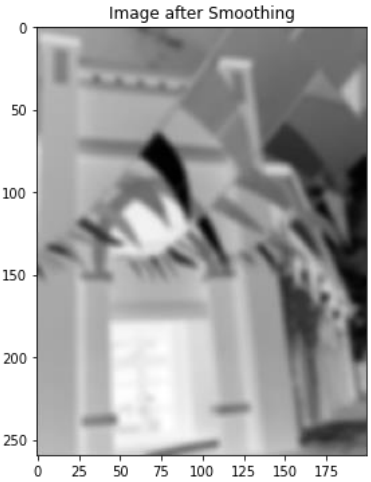
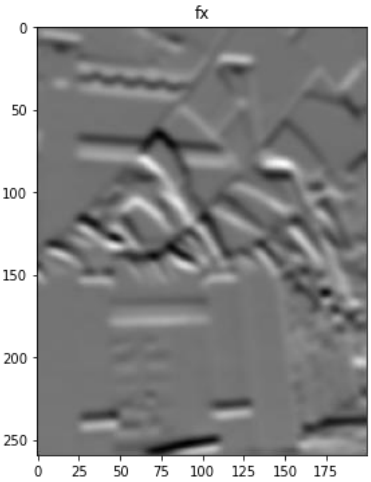
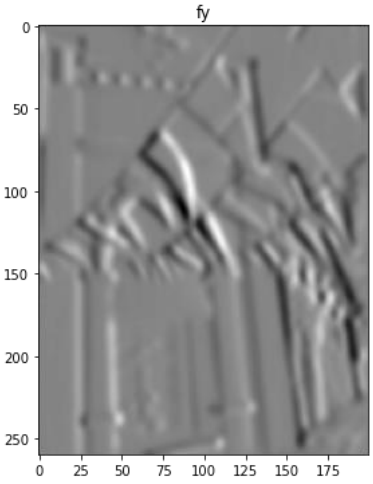
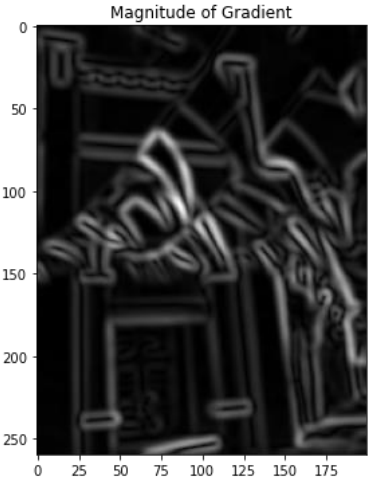
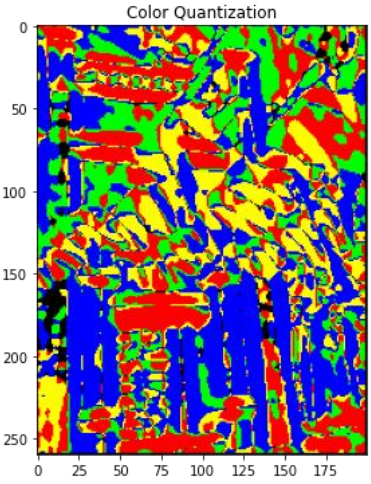


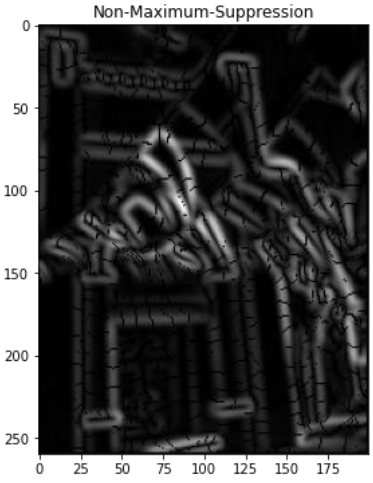
Image after Smoothing

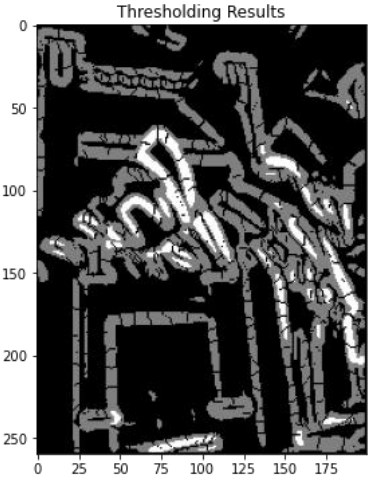

fx


fy


Magnitude of Gradient


Color Quantization


Non-Maximum-Suppression


Thresholding Results

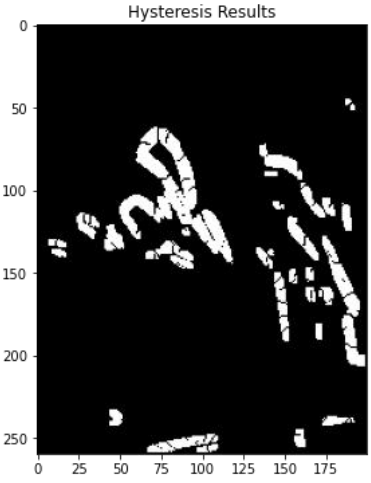
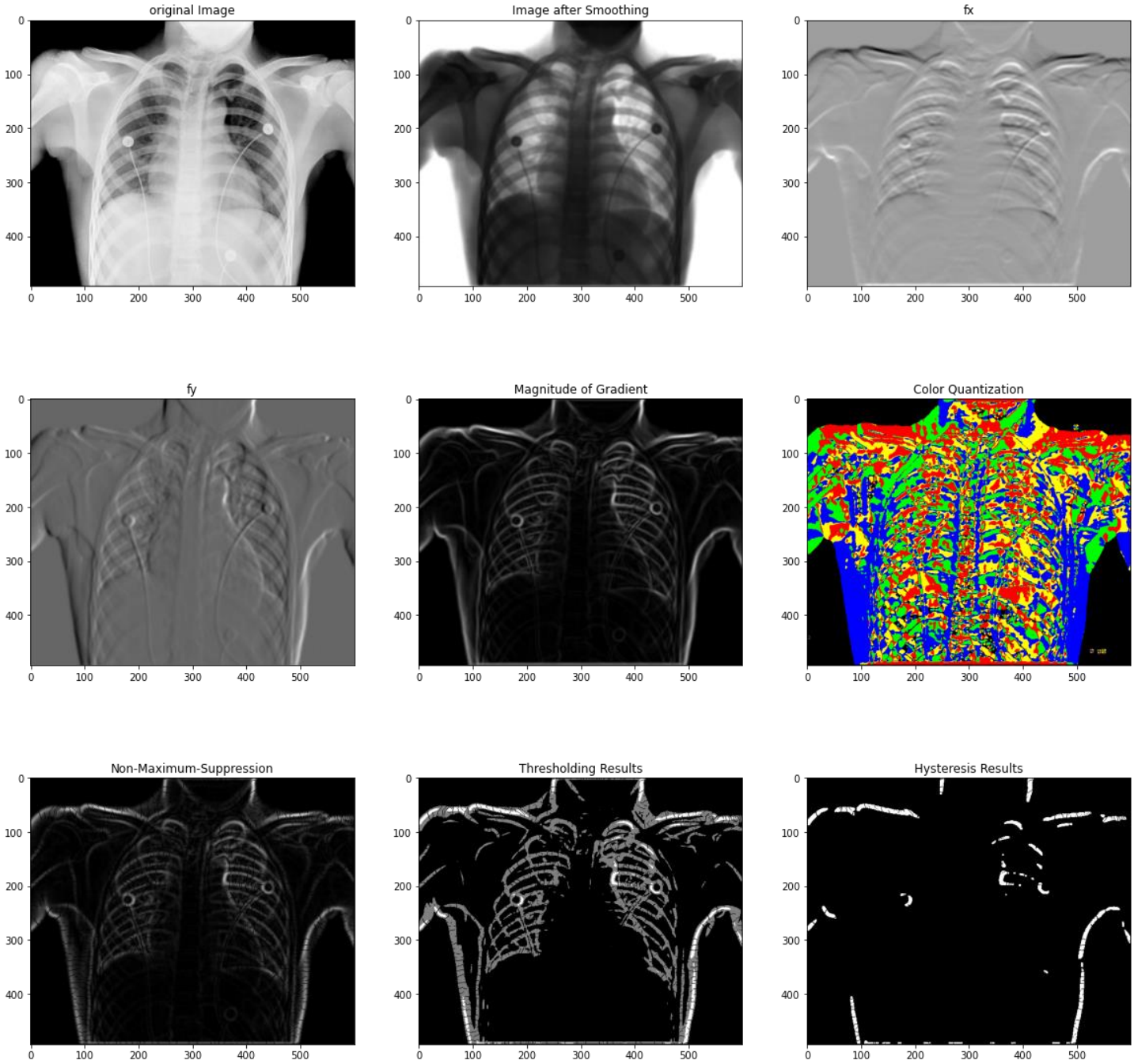

Hysteresis Results

Image: X-ray



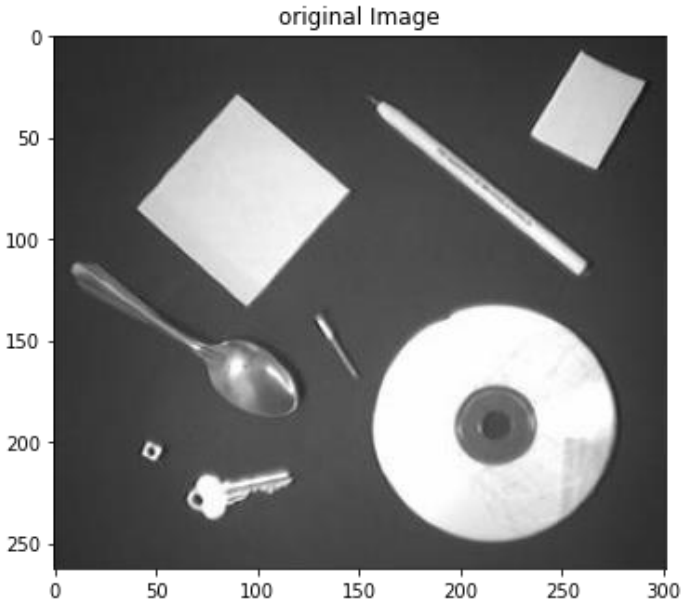
Discussion:

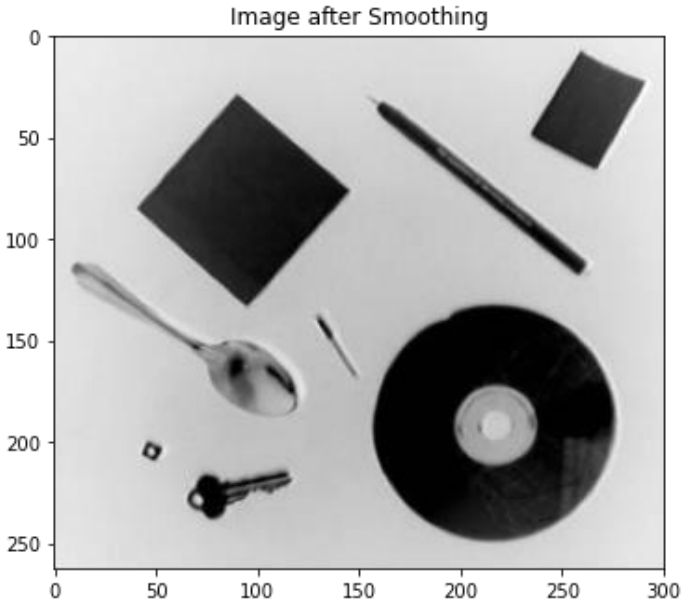
- Above all 3 experiments are performed on images with sigma (0,5,1,2) with HT and LT pair(100,30).
- Vertical and Horizontal edges can be seen by fy and fx respectively.
- Non-Maximum-Suppression is making our edges thin which can be seen from the plot of Non-Maximum-Suppression results.
- The thresholding plot shows both weak edges as gray and strong edges as white.
- Hysteresis Thresholding transforming all weak edges into strong ones according to the specified criteria
- One different effect of the small and larger filter can be seen in images that the edges of the images with large seem to be more smearing out of the pixel value than the 3x3 filter.

Discussing the effect:

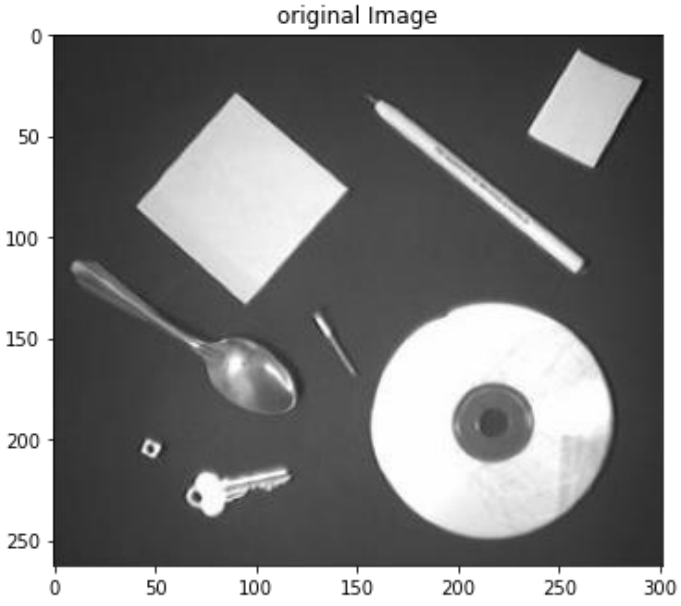
- Increasing the filter size makes the image more blur. As the given example below one image with filter size (3,3), (5,5), and (7,7).

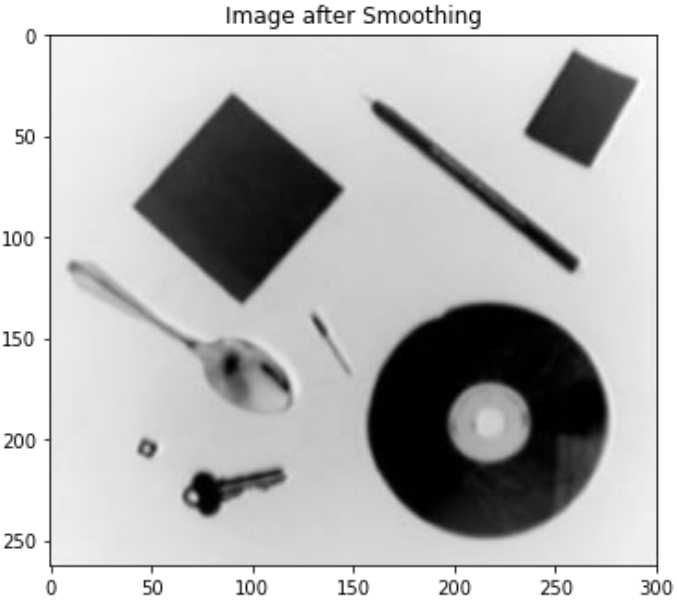
3x3 filter results



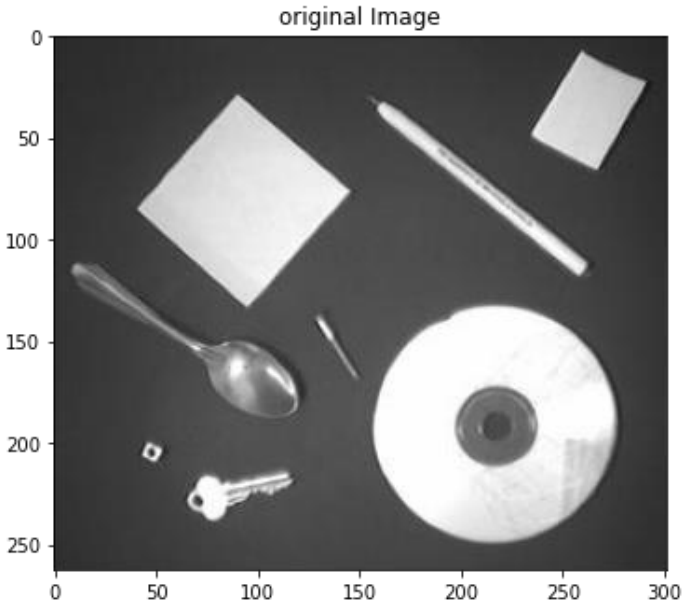


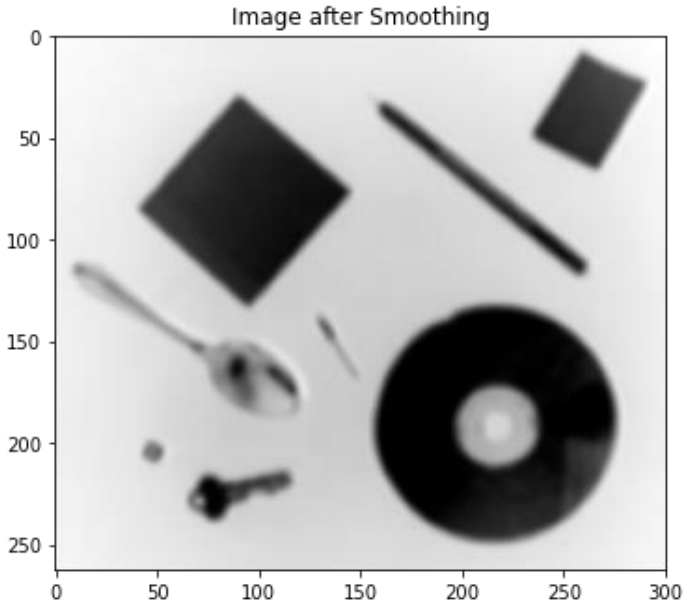
5x5 filter results





7x7 filter results

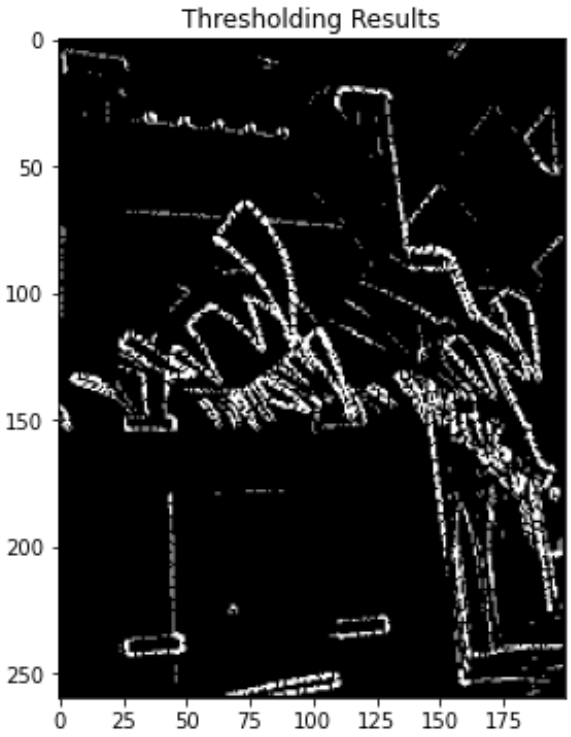


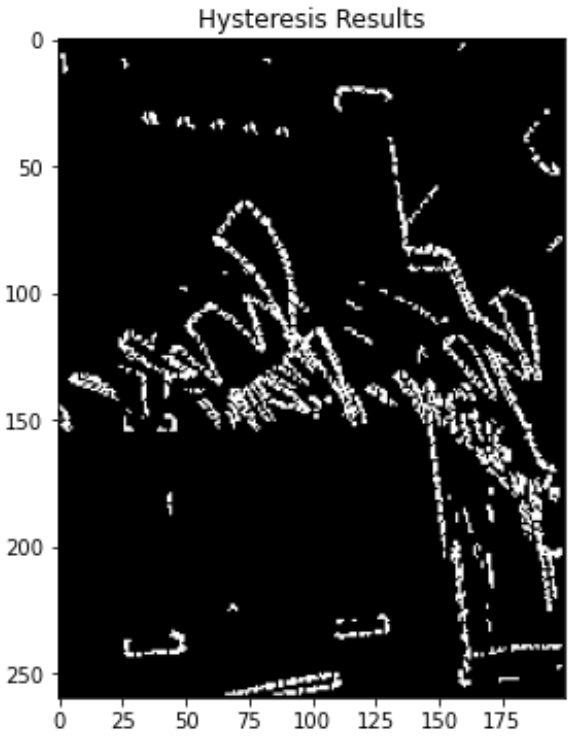


- Blurry images make the localization of edges more difficult because it causes smearing out the value of a given pixel over a larger area of the image.

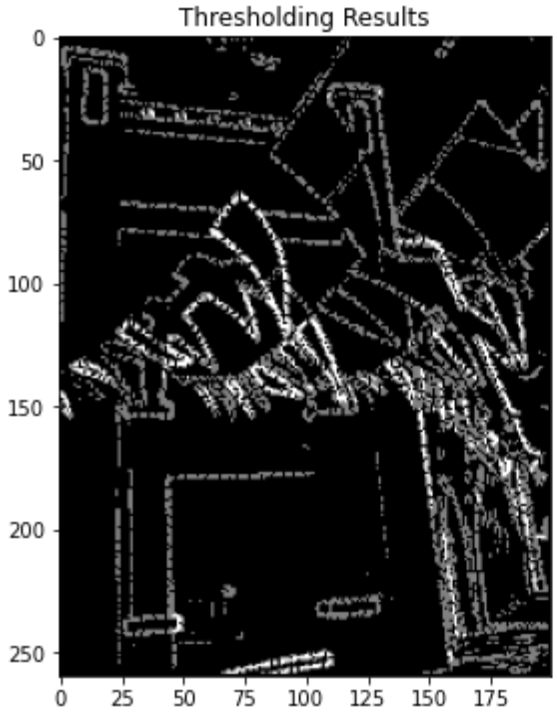
For sigma =1, submit 2 different results for 2 different pairs of Th, Tl in Hysteresis thresholds. And discuss the effect.

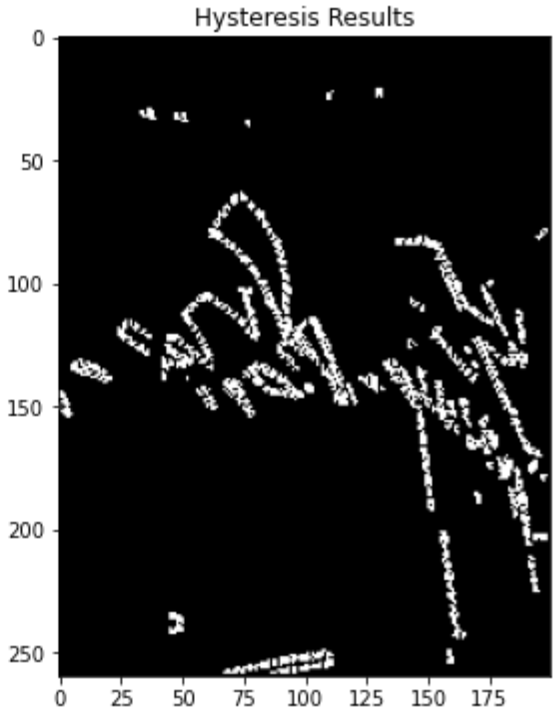
Sigma = 0.5 , HT= 70 , LT = 50



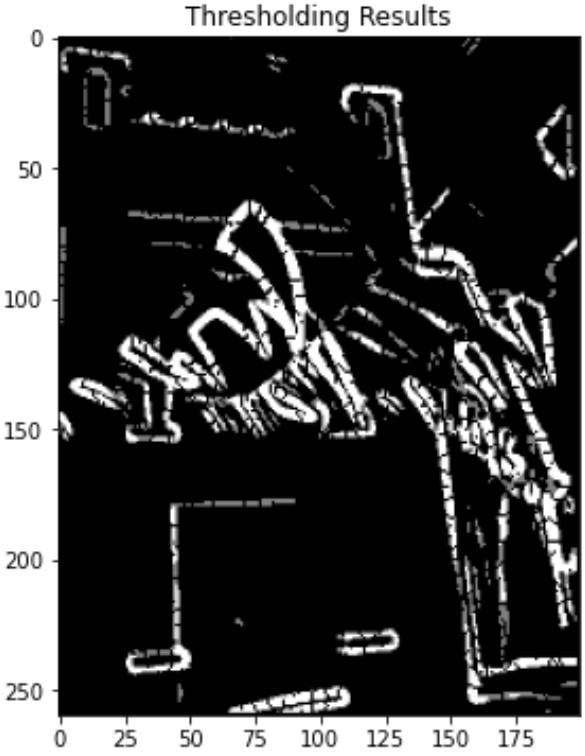


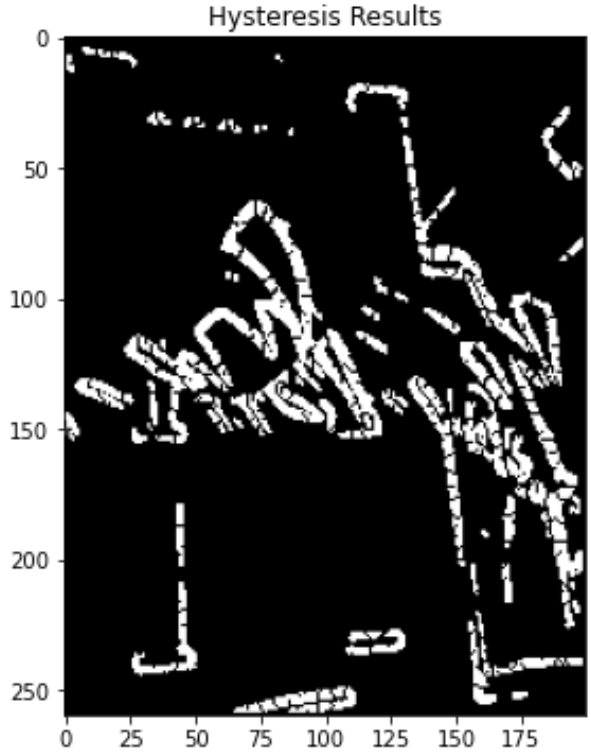
Sigma = 0.5 , HT= 100 , LT = 30



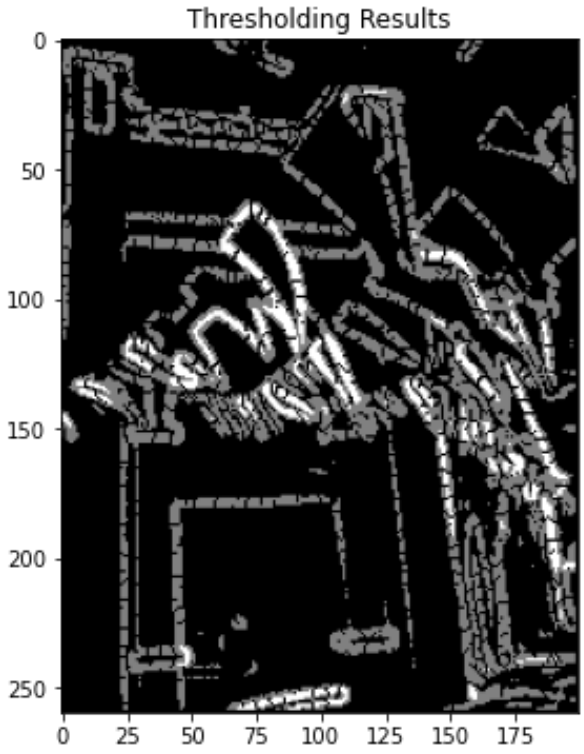


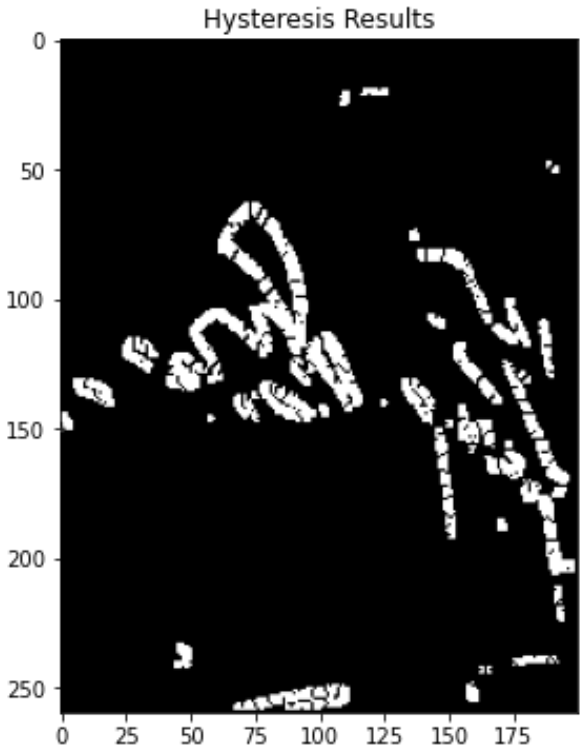
Sigma = 1 , HT= 70 , LT = 50



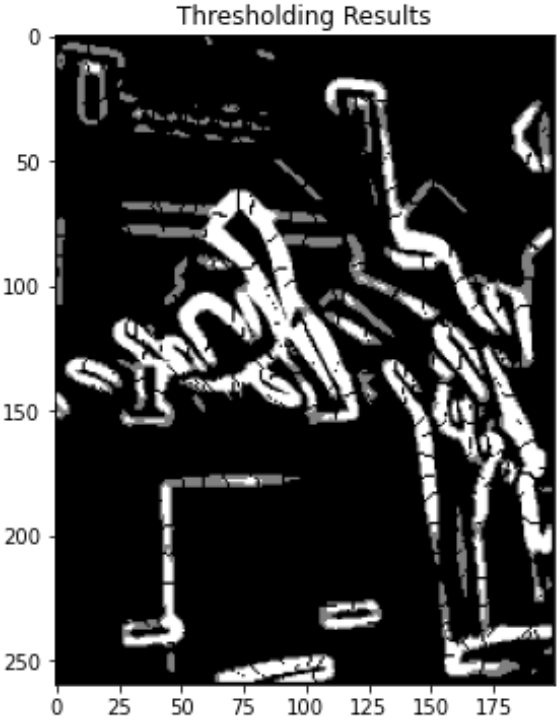


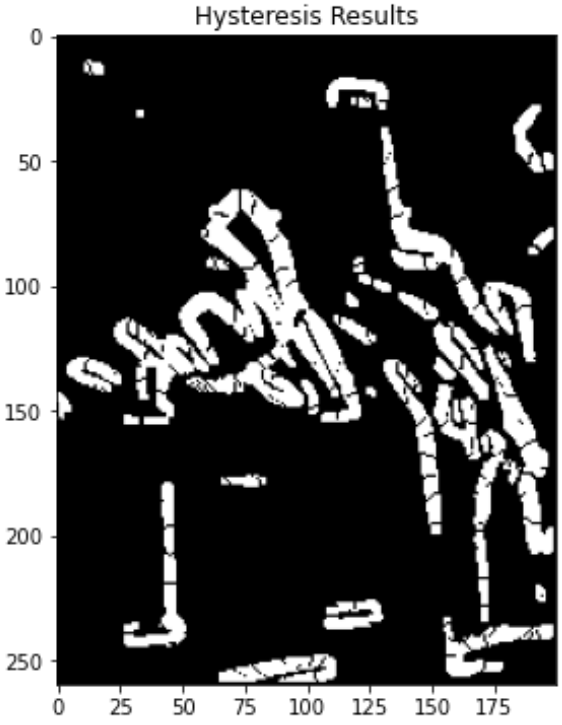
Sigma = 1 , HT= 100 , LT = 30



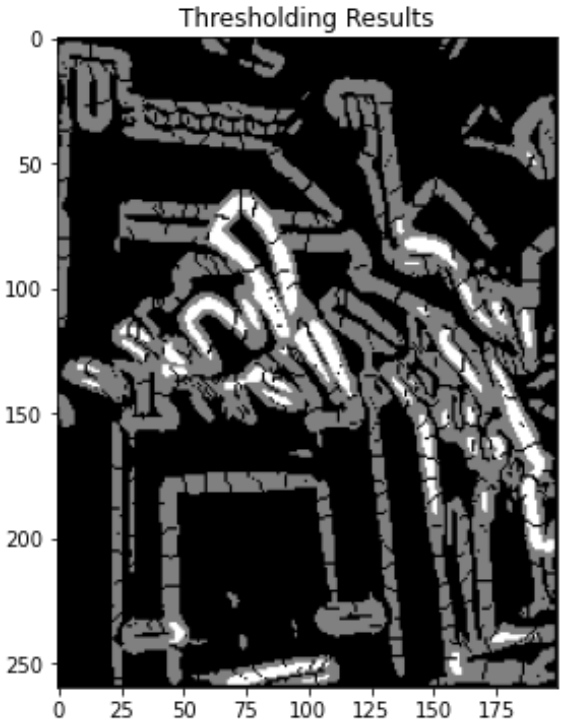


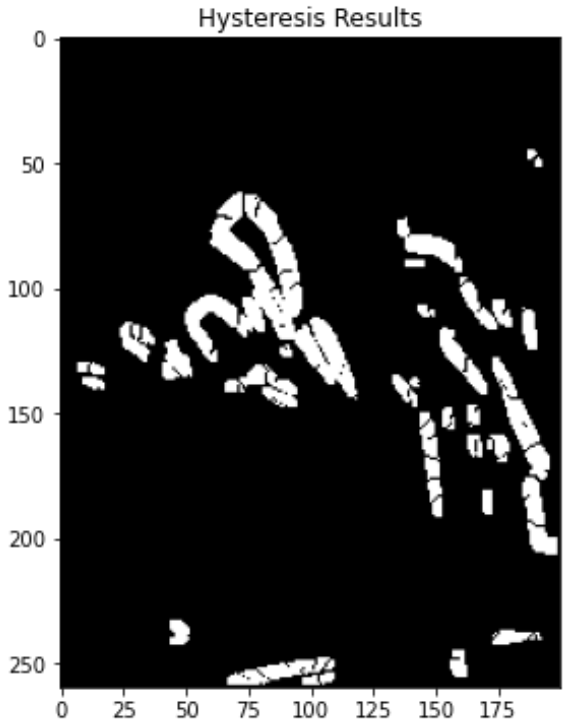
Sigma = 2 , HT= 70 , LT = 50





Sigma = 2 , HT= 100 , LT = 30





- As I have discussed before in the Thresholding results image gray edge presents the weak edges and the white one presents the Strong edges. The hysteresis threshold presents only Strong edges.
- If we compare the Thresholding results of both images the image with (30,100) tends to have more edge information than an image with (50,70) because the minimum threshold is 30 which is smaller than another image threshold. Therefore images with (30,100) threshold setting have more edges than other images.
- If we compare the Hysteresis results, the image (50,70) has more strong edges than the image with (30,100) threshold setting. Most of the edge information is missing in the image with (30,100). This happens because the difference between lower and higher thresholds is big in this Image a difference of 70 while the above image has a difference of 20. The hysteresis transforms the weak pixel into a strong one the pixel if and only if at least one pixel around being processed is a strong one. Due to the larger difference in high threshold and low threshold, the edges information is less than the previous image which has a difference of only 20.
- The only difference we can see in the image is the increasing filter size has made edges smear out of the pixel.

Conclusion:

In this report, the implementation and experiments of canny edge detection have been discussed. The first part of the report discusses the implementation of the report divided into 6 sections with an explanation of how things are working. The second part of experiments has been performed on images with different values of sigma such as 0.5,1,2 and threshold pairs; (50,70) and (30,100). The experiments show that how sigma values affect the detection of edges we came to know that a very large value of sigma tends to make image blur and localization of edges difficult. Different pairs of thresholds also affect the final results of canny edge detection. If there is a large difference between the min and max threshold there will be more edge information because it will contain more weak edges and if the difference is small then there will be less edge information. So it depends on what kind of edge information we want to get from the canny edge detector.