

Computer Vision – Assignment2

Sai Teja Karnati (659365999)

We will be working with 2 filters (3, 5) and different LowThresh, HighThresh values from (0.02,0.09)

I have implemented my own **Convolution, GaussianSmoothing, ImageGradient, NonMaximaSuppress, Thresholding, Edge Linking** functions.

I used numpy, cv2, rescale_intensity and math libraries for my work.

I created a **GaussianSmoothing** () function which creates Gaussian kernels based on given parameters and outputs the Convolution (Image, GKernel). We use these differently smoothed images for edge detection.

We will explore results of Lena and Test of assignment with various parameters. I have written my observation beneath each part.

Algorithm of Canny edge Detector:

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

First, we get our **Gaussian Smoothed** image using the GaussianSmoothing function implemented with given parameters. Then we calculate XGradient and YGradient using Sobel filter as explained in class. We get Gradient image and theta value from here.

Then, we use **Non-Maximum suppression** , an edge thinning technique to suppress the edge strength to 0 if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction.

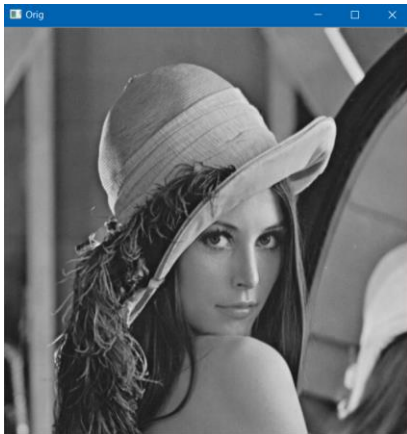
Then, we remove edge pixels with weak gradient value given by our low threshold and high threshold parameters. Then we apply edge linking to obtain connected edges.

Results with Image Lena

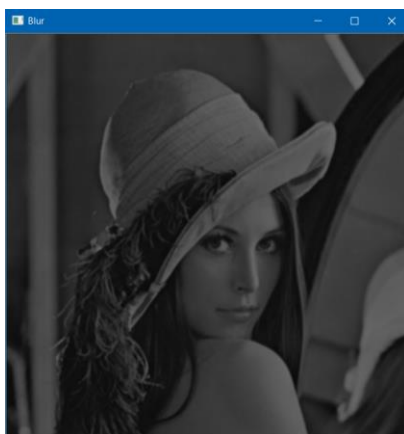
Kernel 3

Lena Image (Kernel =3, 1.5 , LowThresh, HighThresh= 0.02,0.1)

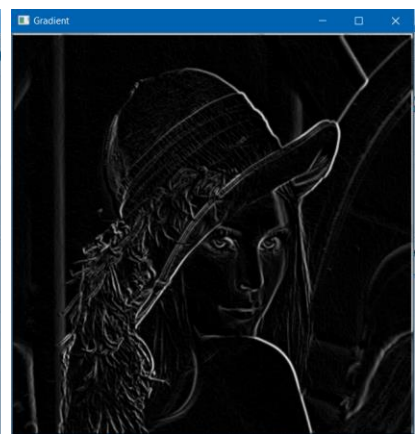
Original Image



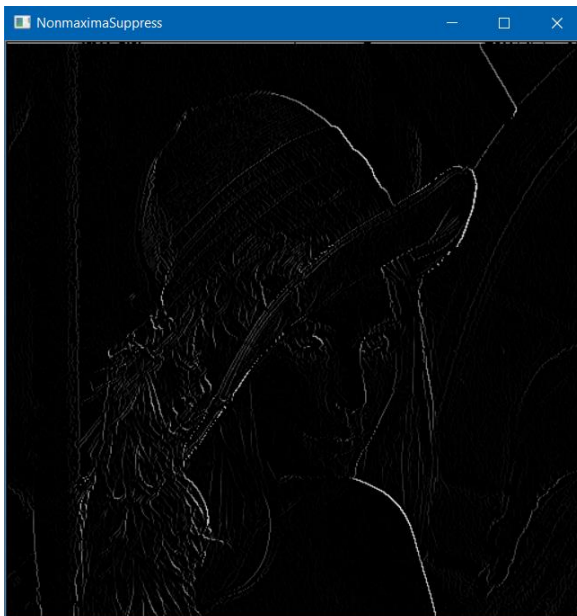
Gaussian Smoothed



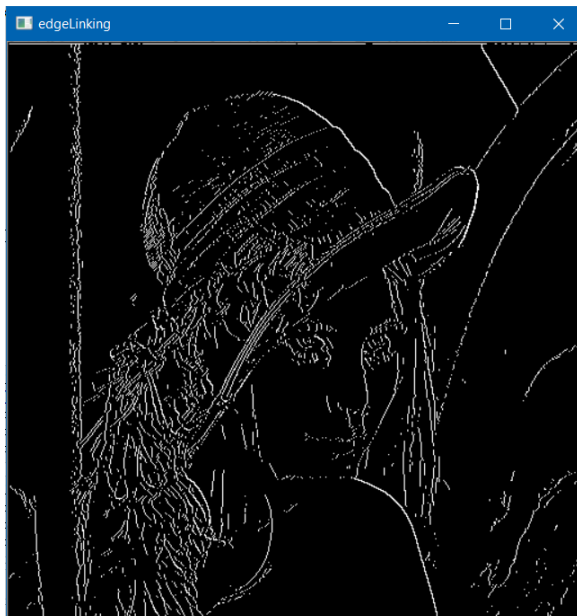
Gradient Image



Suppressed Non maxima image



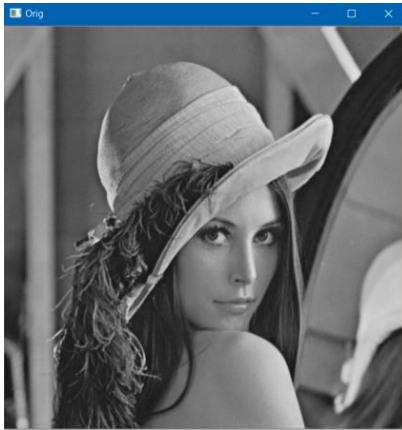
Edge Linking Image



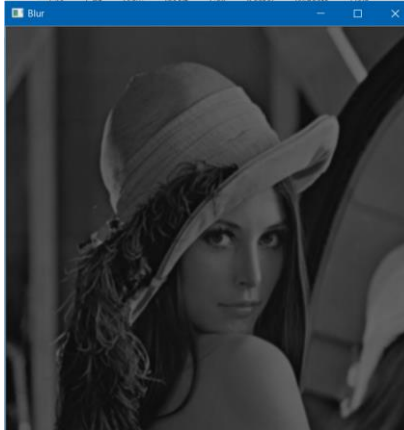
Kernel 3

Lena Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.01,0.05)

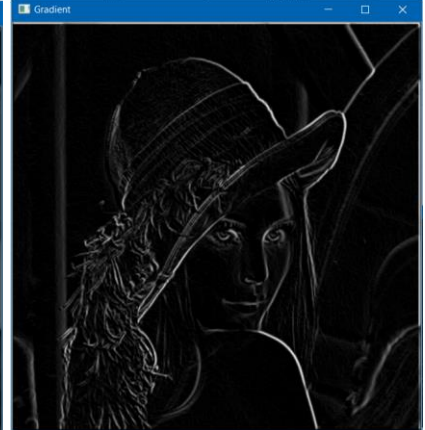
Original Image



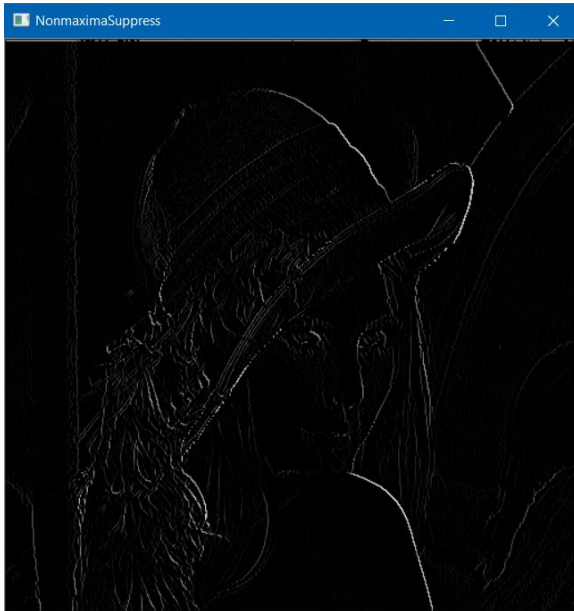
Gaussian Smoothed



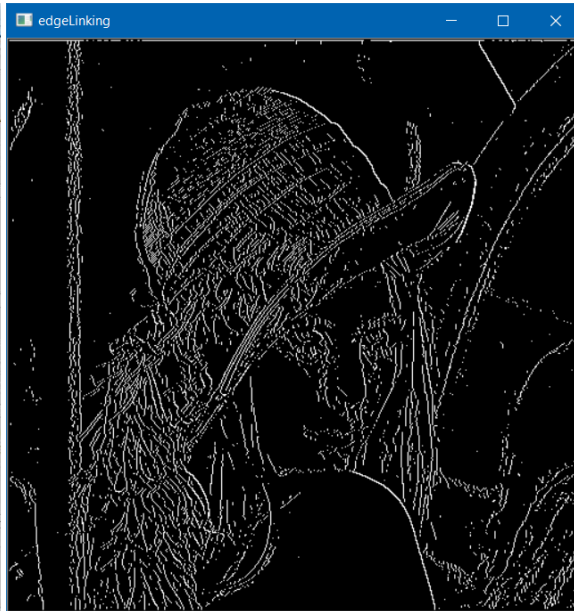
Gradient Image



Suppressed Non maxima image



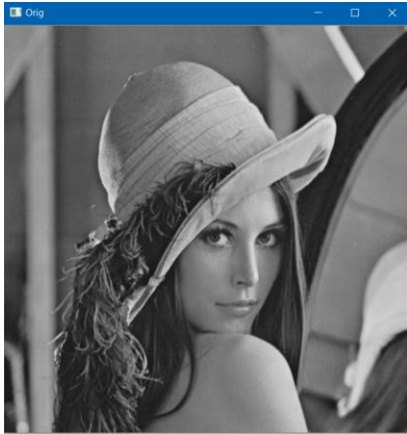
Edge Linking Image



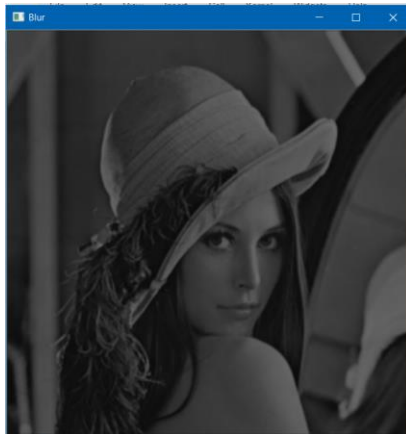
Kernel 3

Lena Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.05,0.1)

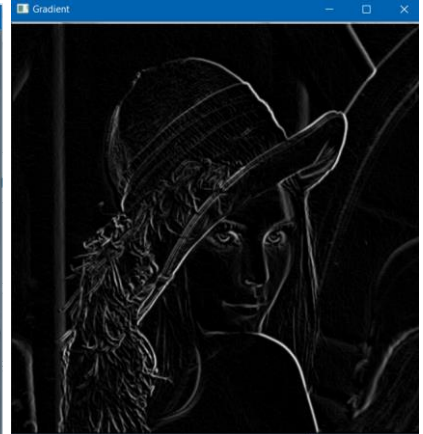
Original Image



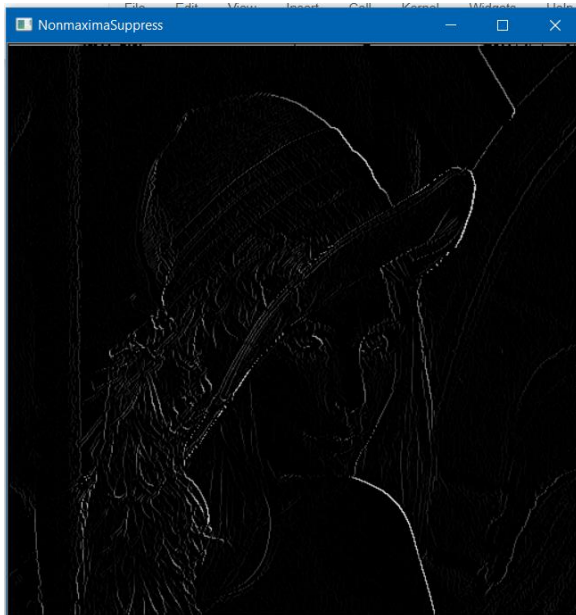
Gaussian Smoothed



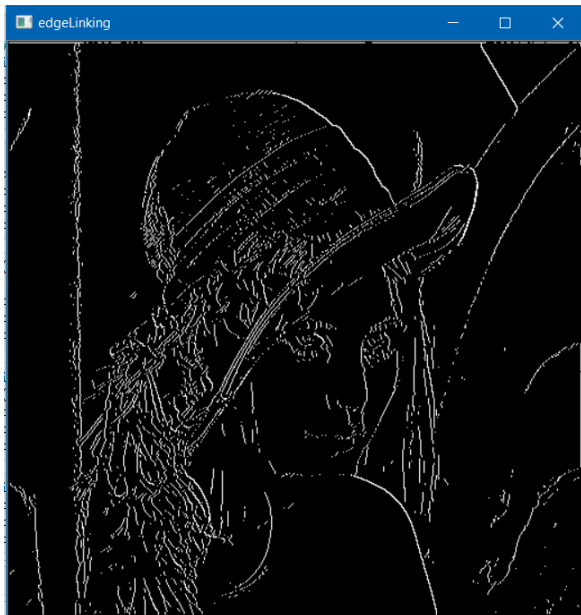
Gradient Image



Suppressed Non maxima image



Edge Linking Image



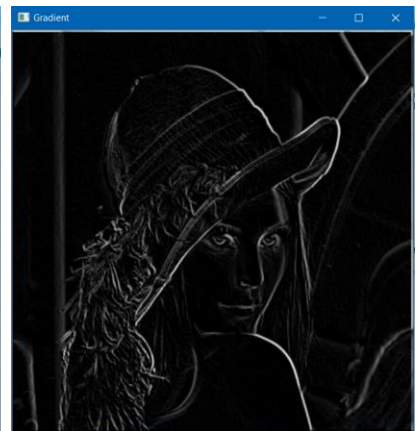
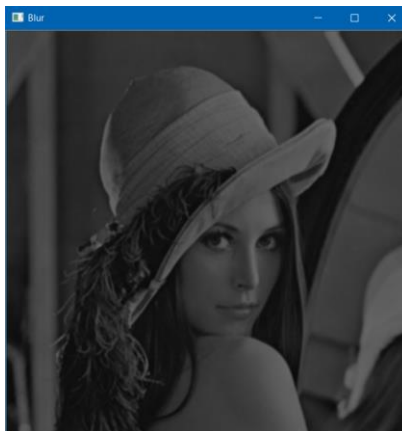
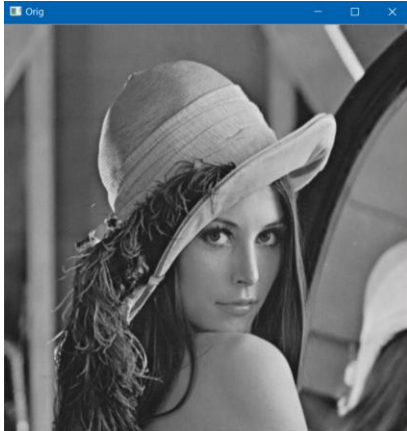
Kernel 3

Lena Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.4,0.2)

Original Image

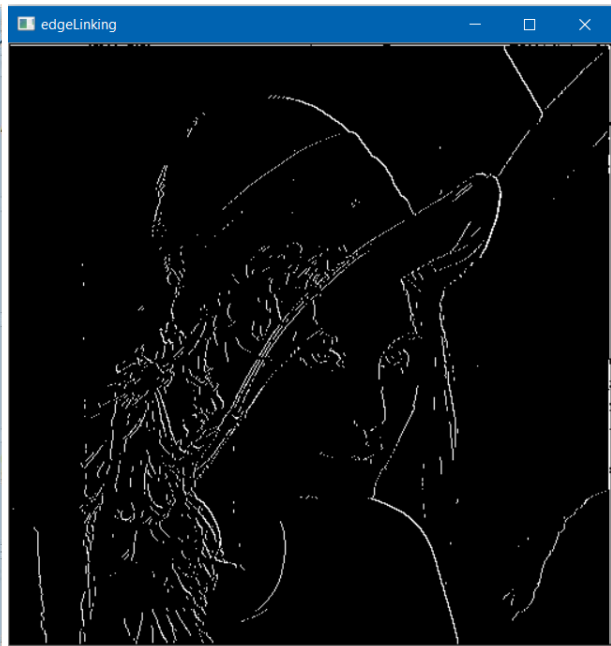
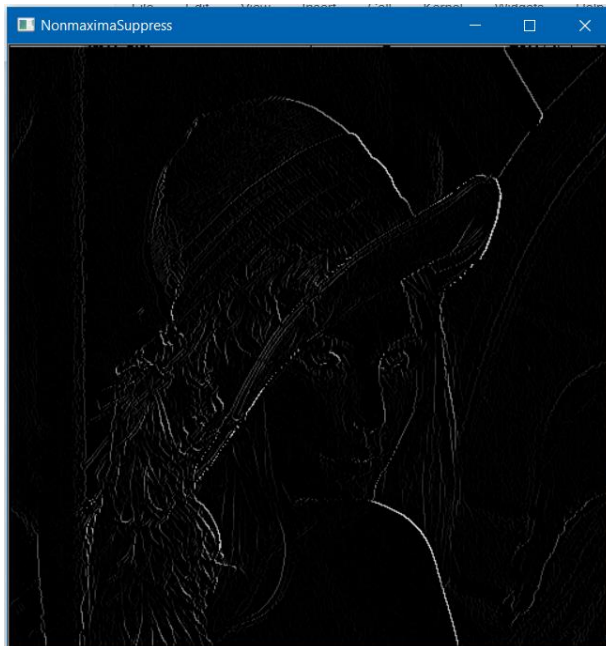
Gaussian Smoothed

Gradient Image



Suppressed Non maxima image

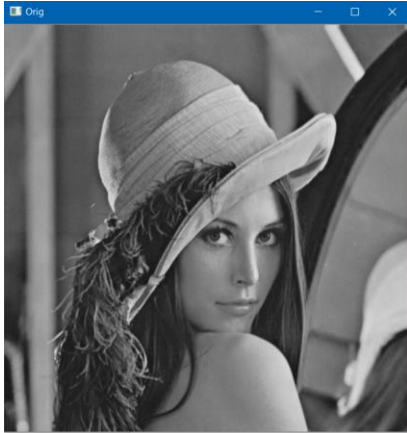
Edge Linking Image



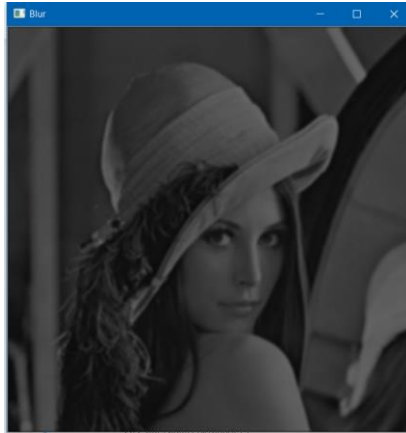
Kernel 5

Lena Image (Kernel =5,2.5, LowThresh, HighThresh= 0.02,0.1)

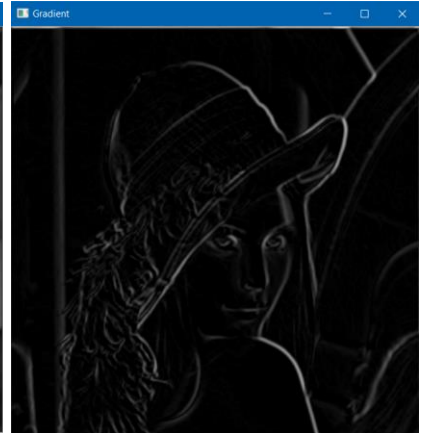
Original Image



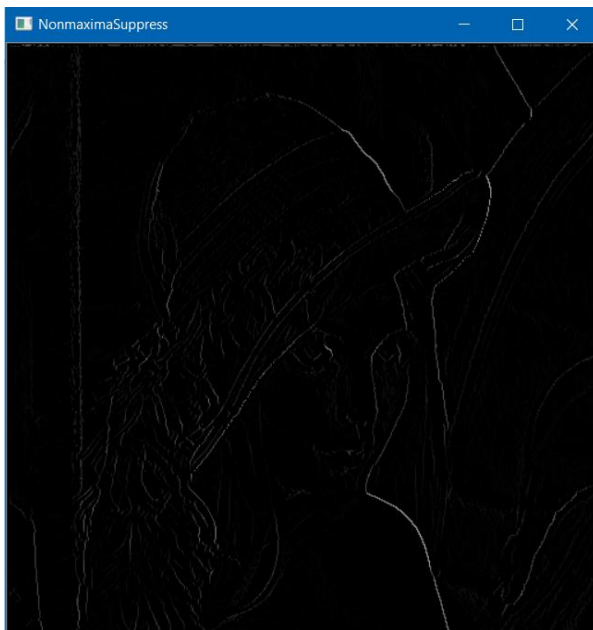
Gaussian Smoothed



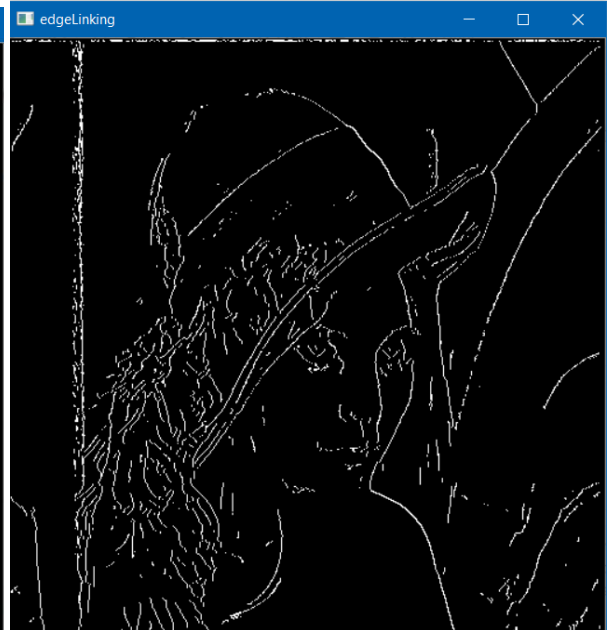
Gradient Image



Suppressed Non maxima image



Edge Linking Image



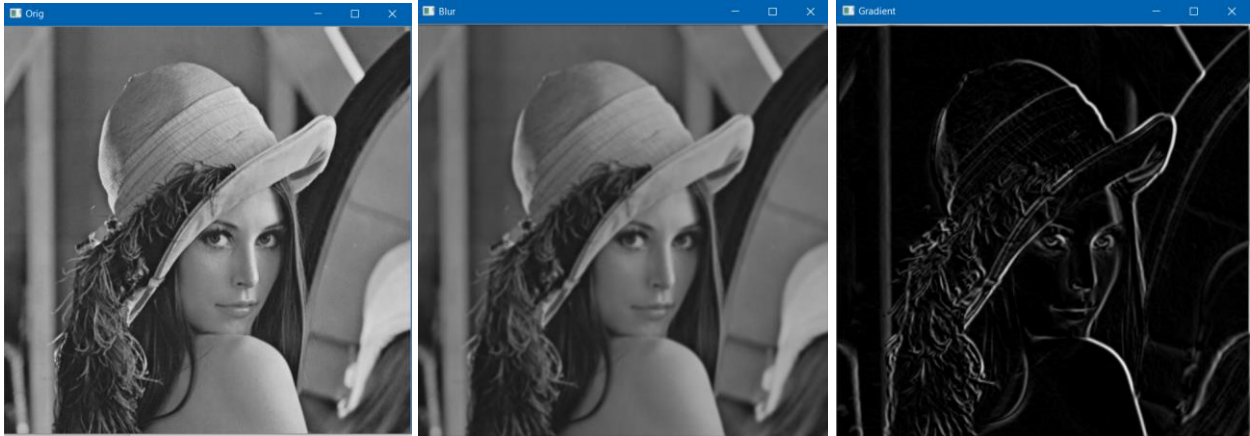
Kernel 5

Lena Image (Kernel =5,1.5, LowThresh, HighThresh= 0.02,0.1)

Original Image

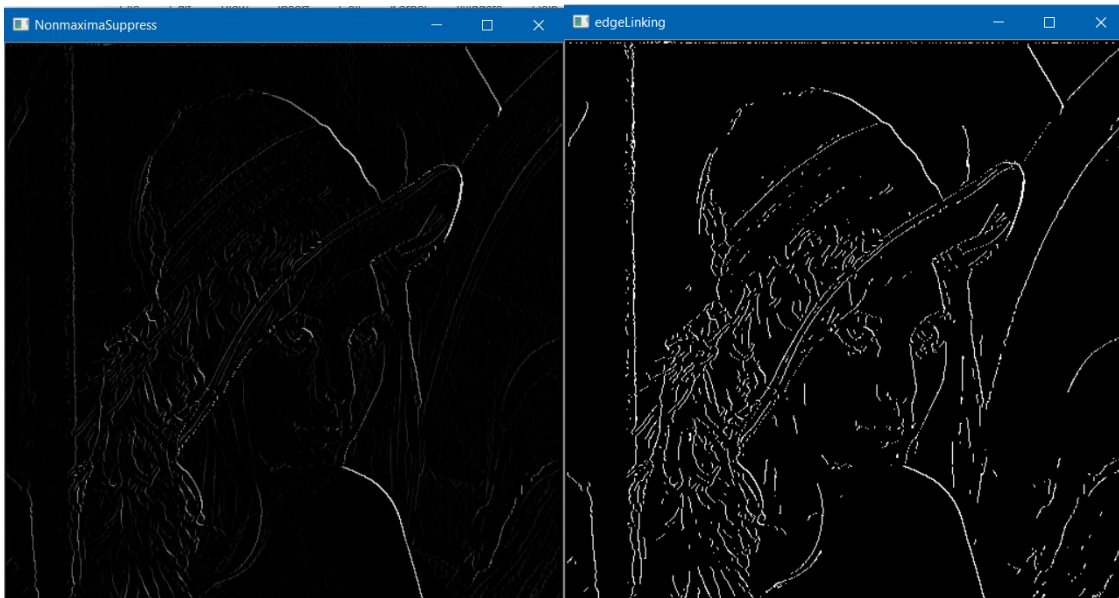
Gaussian Smoothed

Gradient Image



Suppressed Non maxima image

Edge Linking Image



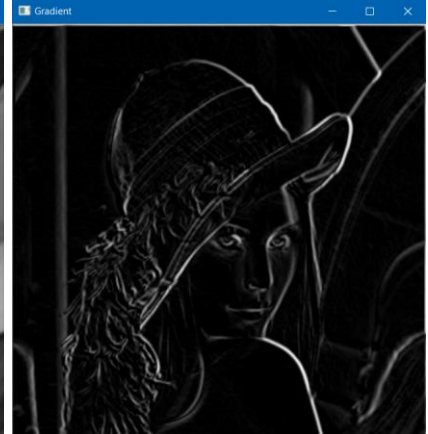
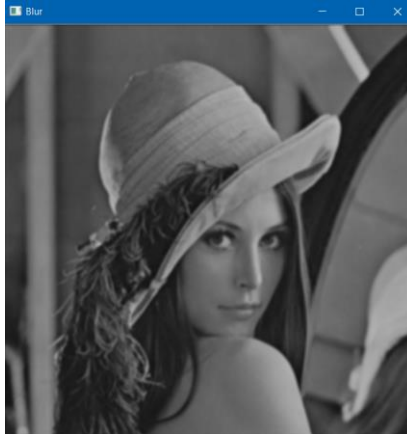
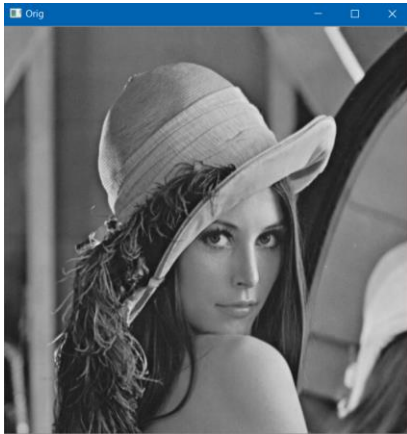
Kernel 5

Lena Image (Kernel =5, 1.5, LowThresh, HighThresh= 0.03,0.12)

Original Image

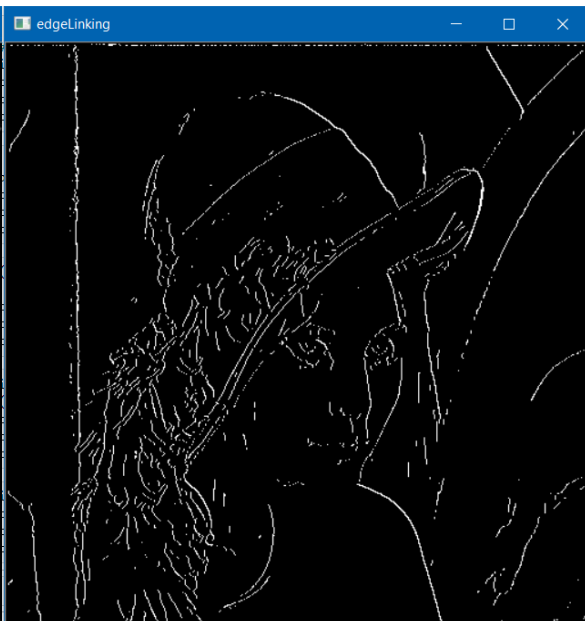
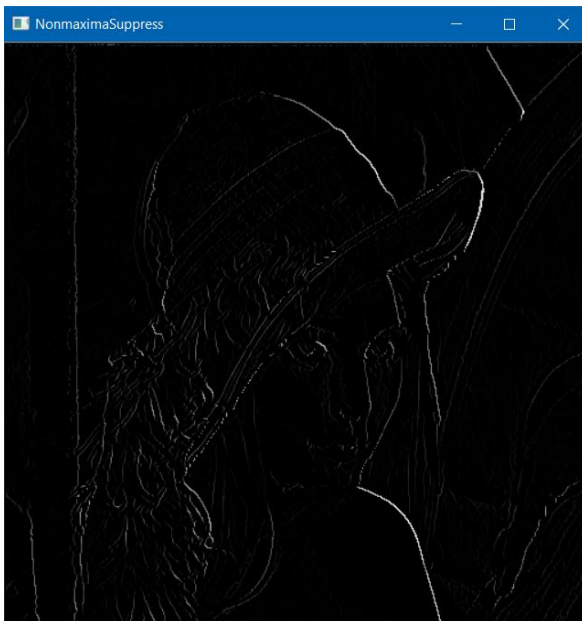
Gaussian Smoothed

Gradient Image



Suppressed Non maxima image

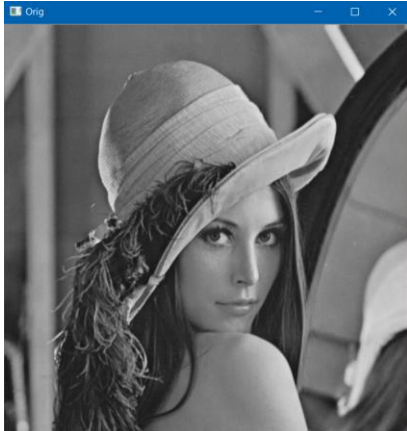
Edge Linking Image



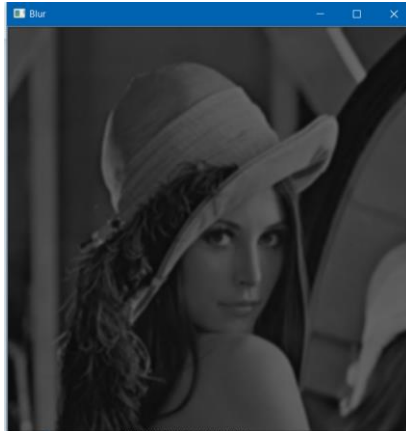
Kernel 5

Lena Image (Kernel =5,2.5, LowThresh, HighThresh= 0.03,0.12)

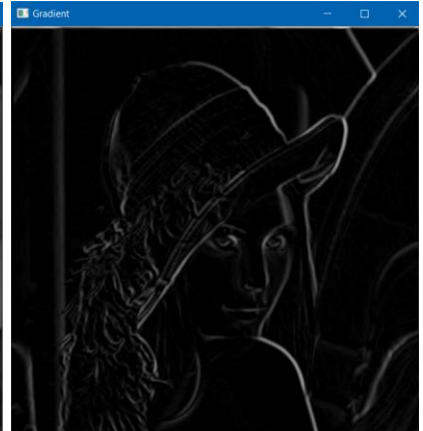
Original Image



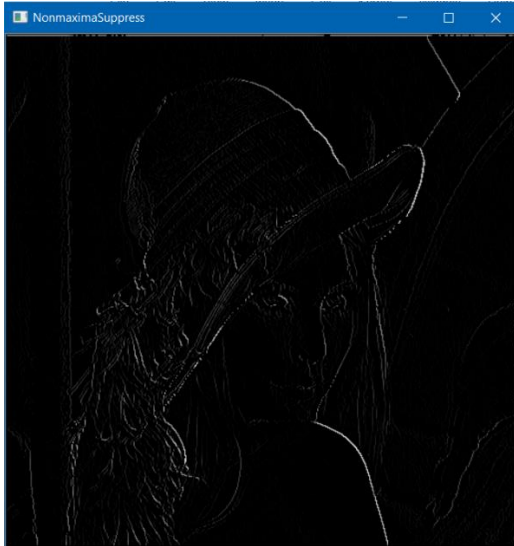
Gaussian Smoothed



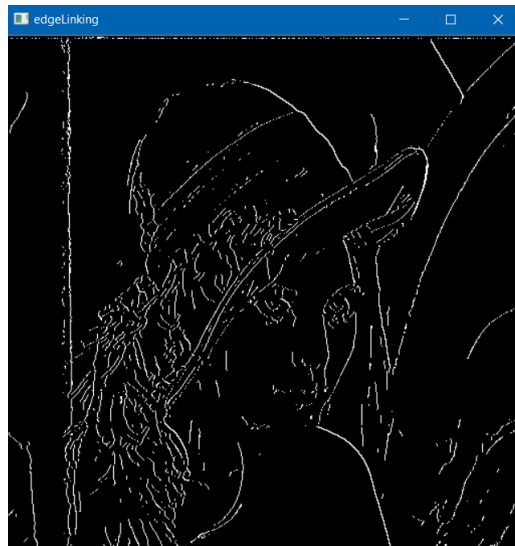
Gradient Image



Suppressed Non maxima image

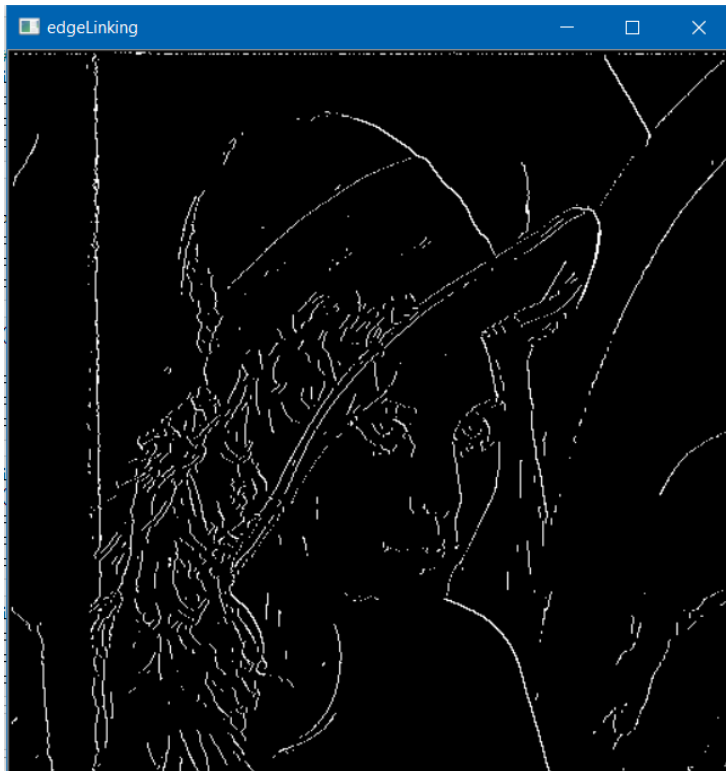


Edge Linking Image



Based on our observations: My optimal parameter choice is:

For lena image (Kernel =(5,1.5) ; LowThresh, HighThresh= 0.03,0.12)



Observations:

Observations:

Kernels: Compared to 5 kernel, 3 kernel is giving a more cluttered edge link images. I think it is because the 3 kernel is catching a lot of small details like the edges in the hat. 5 kernel is ignoring unobservable edges and has clear lines.

We are losing a lot of info in the final result with HighThresh>0.1 and catching too much info for LowThresh <0.02

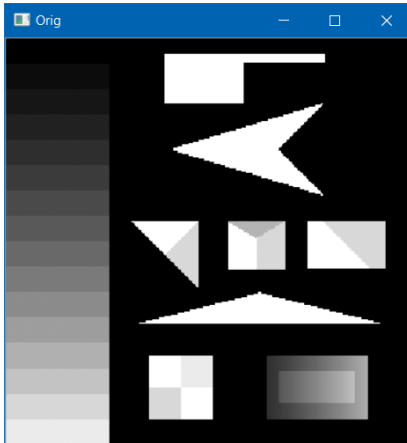
Therefore, our choice of LowThresh, HighThresh= (0.03, 0.12) and kernel = (5, 1.5)

Results with Image Test

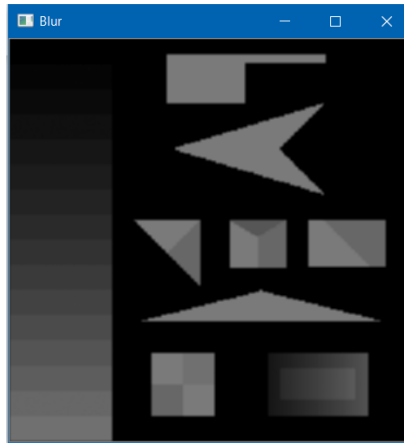
Kernel 3

Test Image (Kernel =3, 1.5, LowThresh, HighThresh= 0.02,0.1)

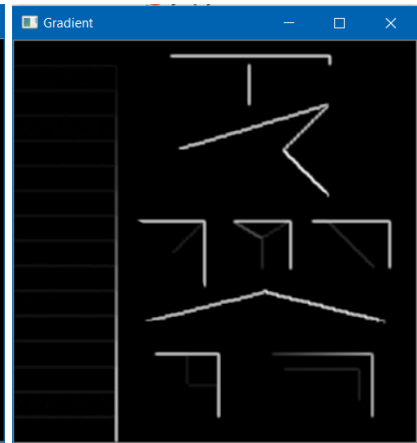
Original Image



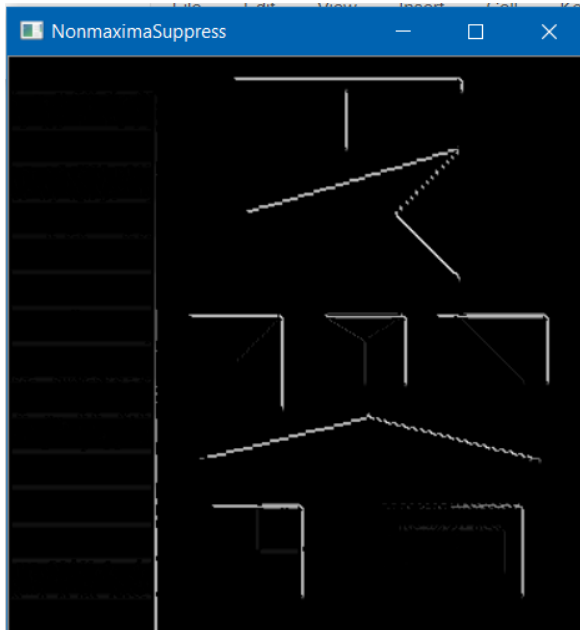
Gaussian Smoothed



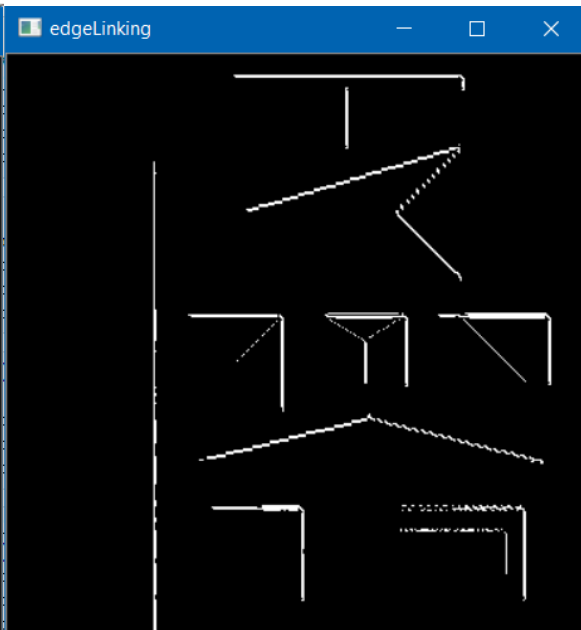
Gradient Image



Suppressed Non maxima image



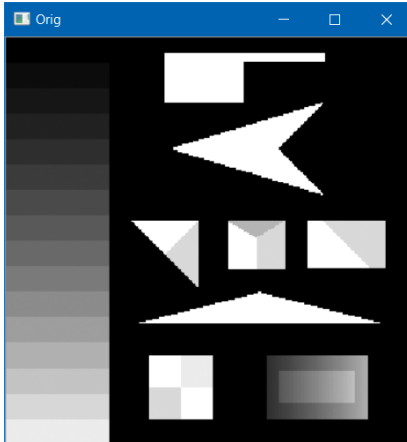
Edge Linking Image



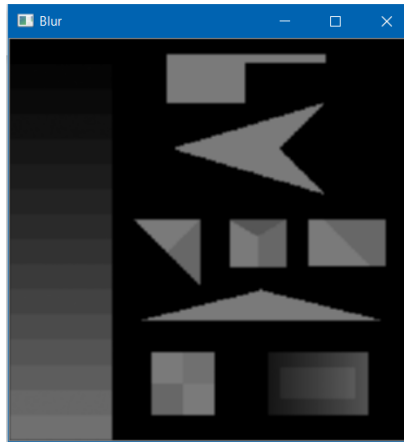
Kernel 3

Test Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.01,0.05)

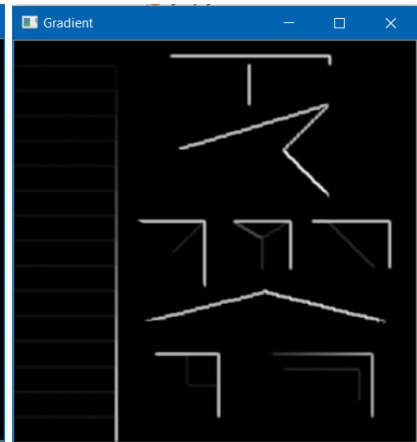
Original Image



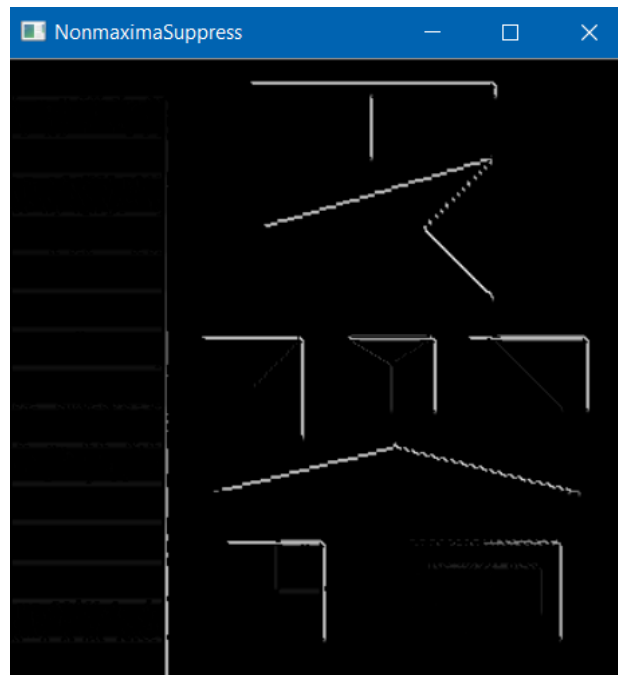
Gaussian Smoothed



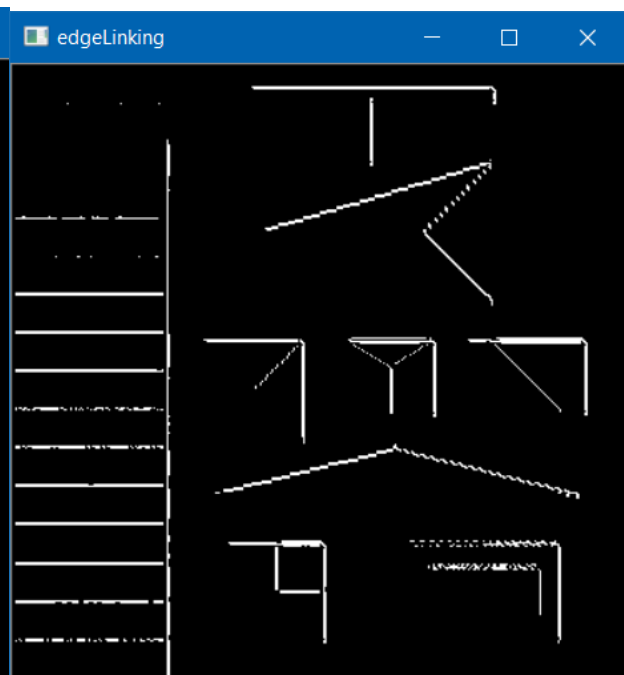
Gradient Image



Suppressed Non maxima image



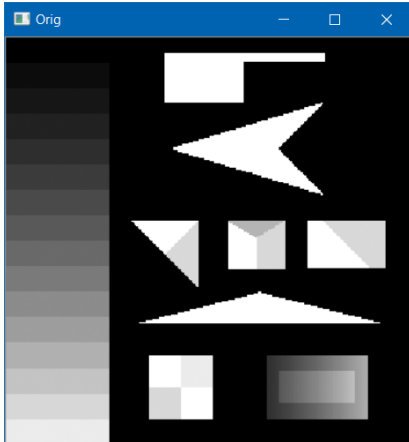
Edge Linking Image



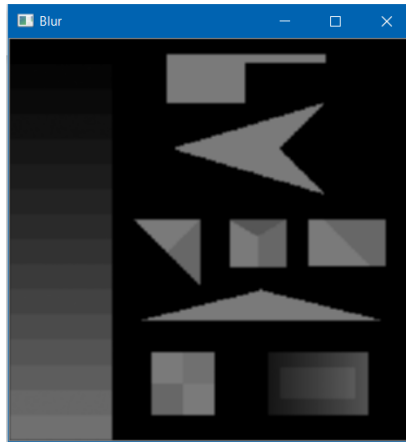
Kernel 3

Test Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.05,0.01)

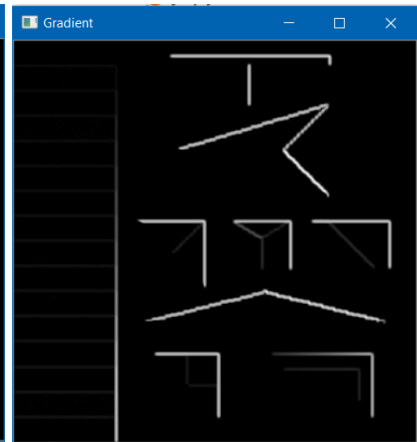
Original Image



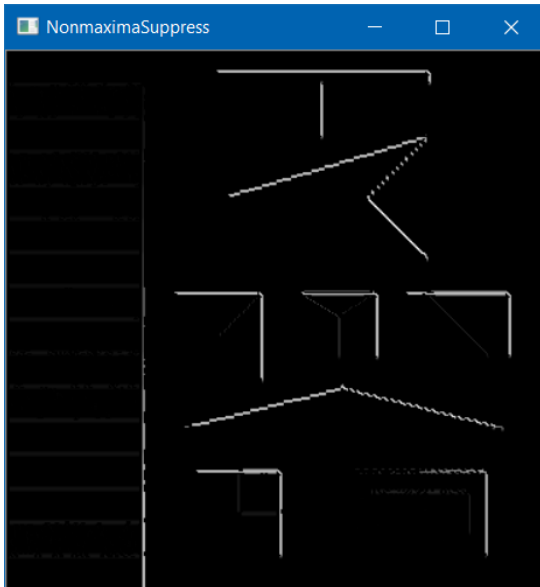
Gaussian Smoothed



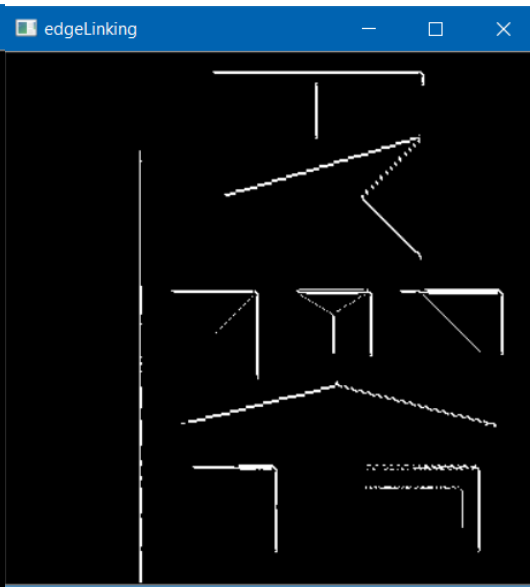
Gradient Image



Suppressed Non maxima image



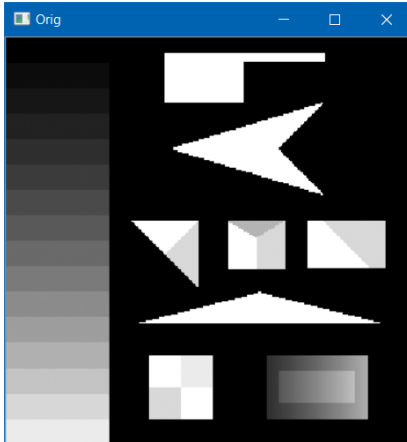
Edge Linking Image



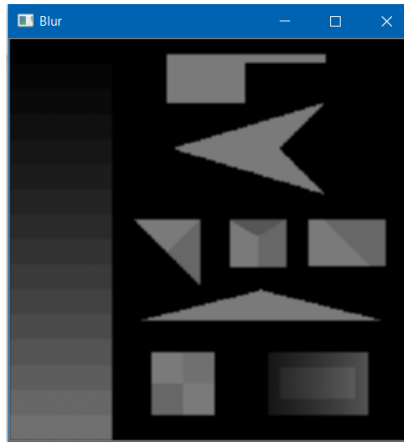
Kernel 3

Test Image (Kernel =3,1.5 , LowThresh, HighThresh= 0.04,0.2)

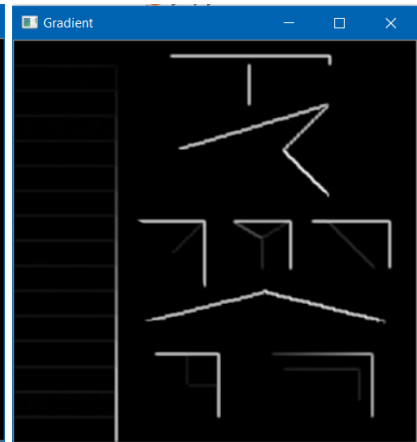
Original Image



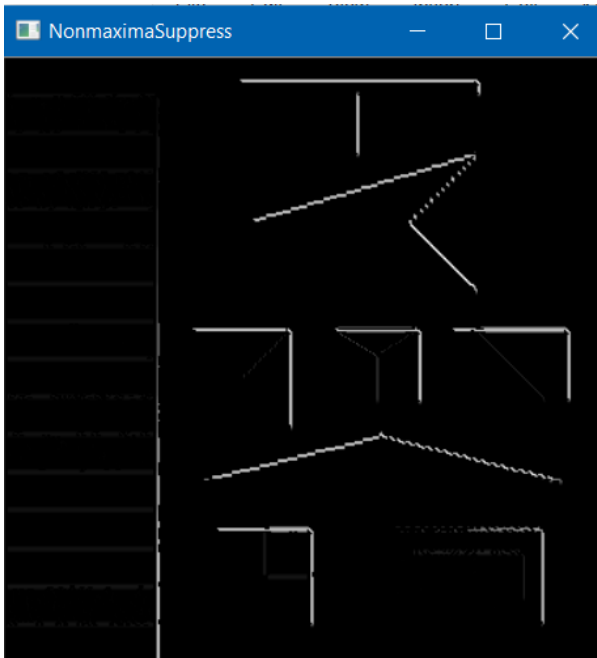
Gaussian Smoothed



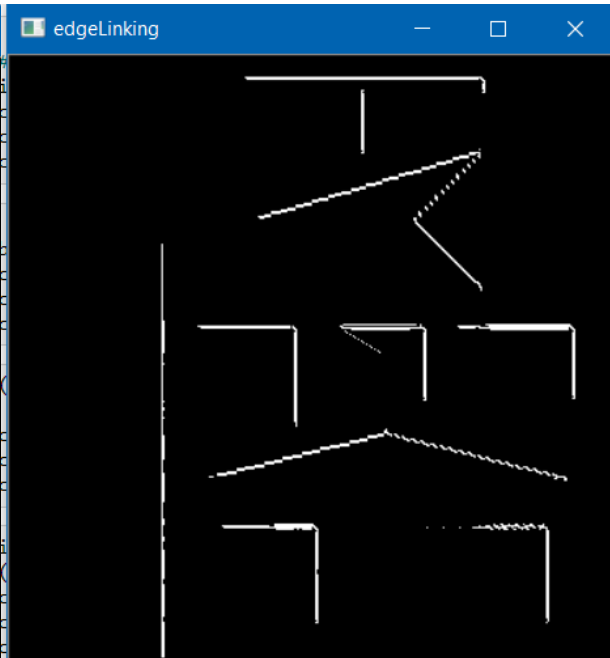
Gradient Image



Suppressed Non maxima image



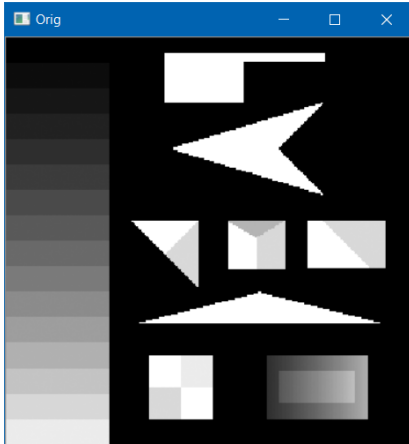
Edge Linking Image



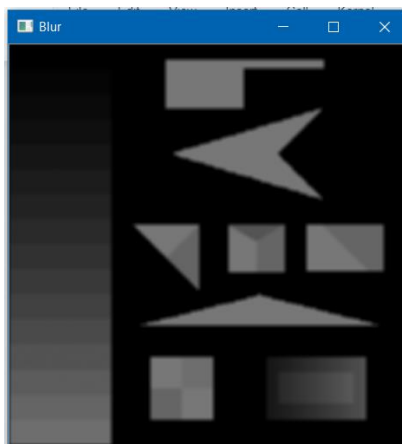
Kernel 5

Test Image (Kernel =5,2.5, LowThresh, HighThresh= 0.02,0.1)

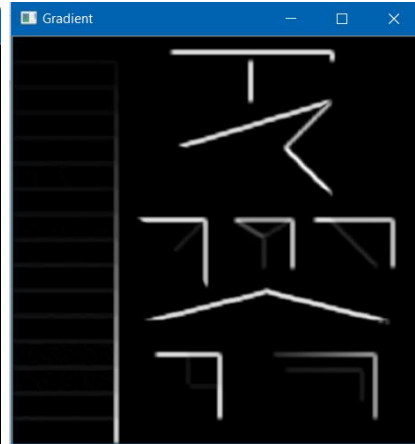
Original Image



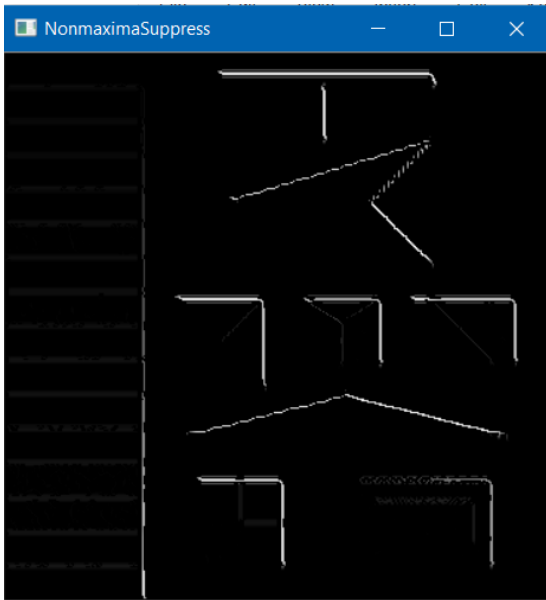
Gaussian Smoothed



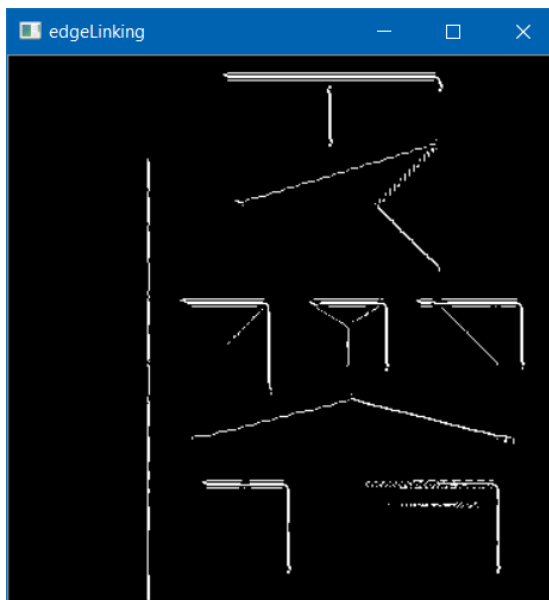
Gradient Image



Suppressed Non maxima image



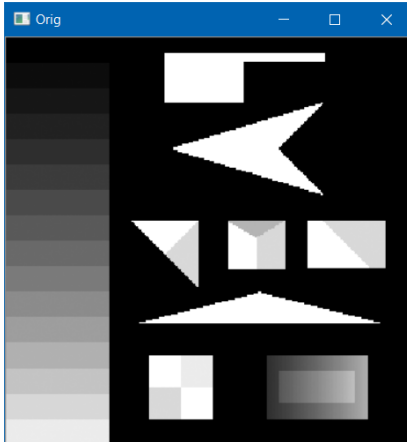
Edge Linking Image



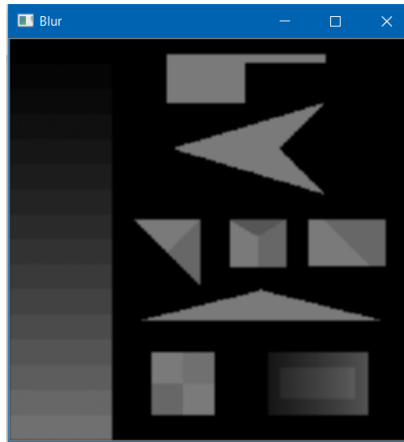
Kernel 5

Test Image (Kernel =5,2.5, LowThresh, HighThresh= 0.01,0.05)

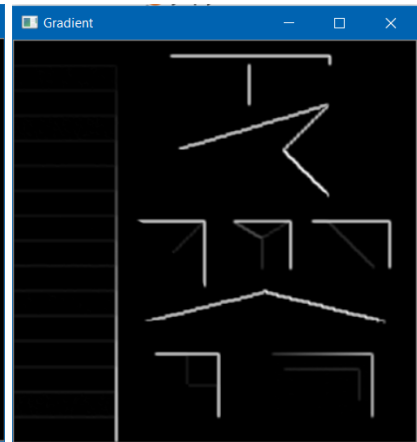
Original Image



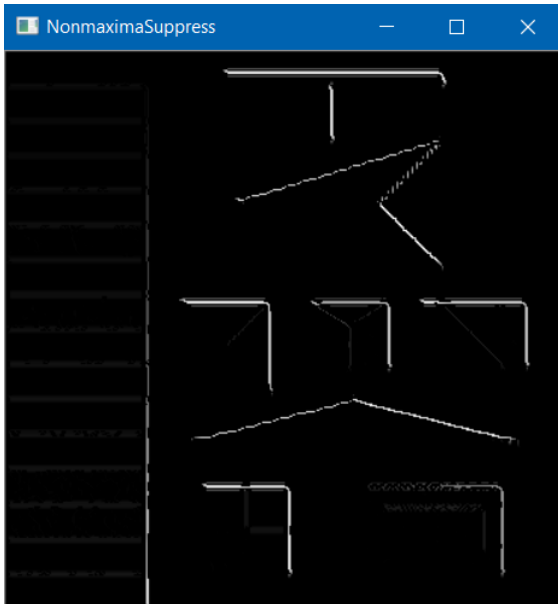
Gaussian Smoothed



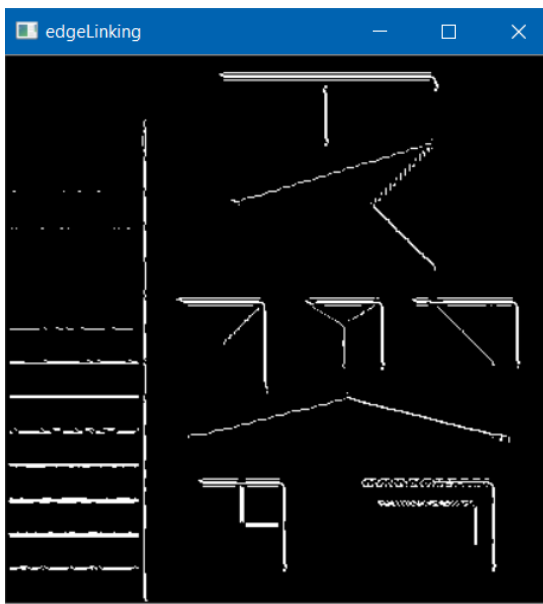
Gradient Image



Suppressed Non maxima image



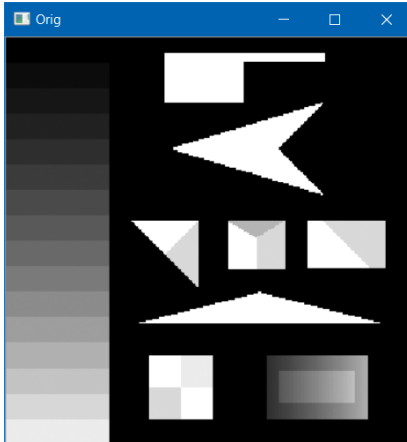
Edge Linking Image



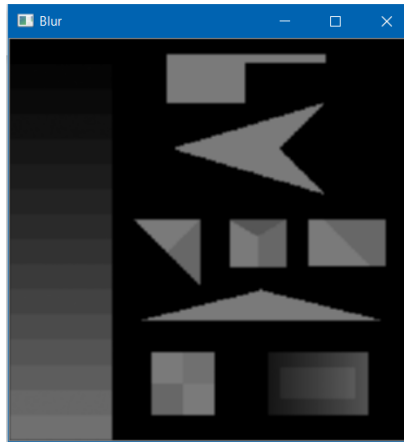
Kernel 5

Test Image (Kernel =5,2.5, LowThresh, HighThresh= 0.05,0.1)

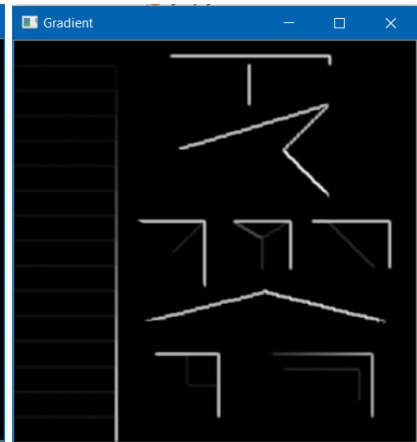
Original Image



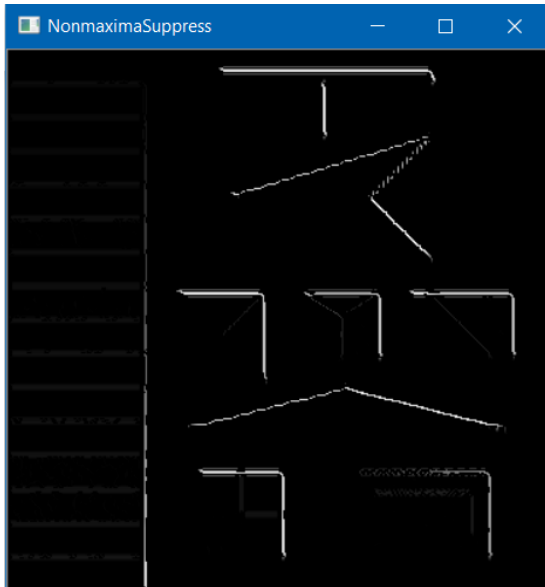
Gaussian Smoothed



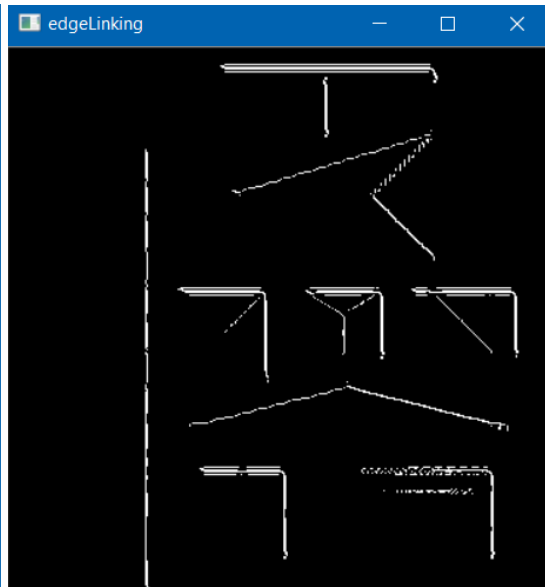
Gradient Image



Suppressed Non maxima image



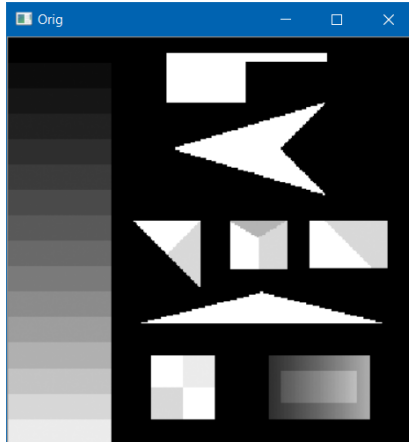
Edge Linking Image



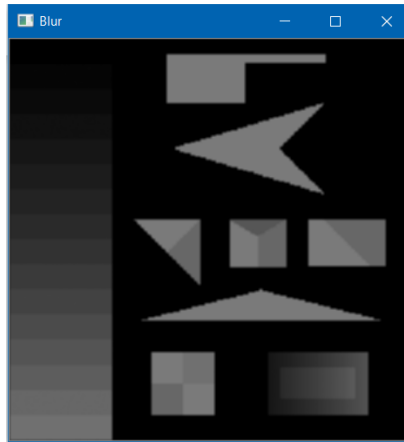
Kernel 5

Test Image (Kernel =5,2.5 , LowThresh, HighThresh= 0.04,0.2)

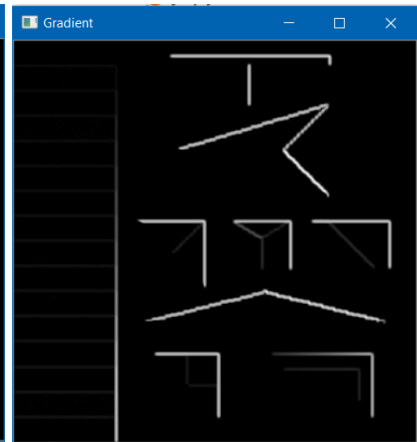
Original Image



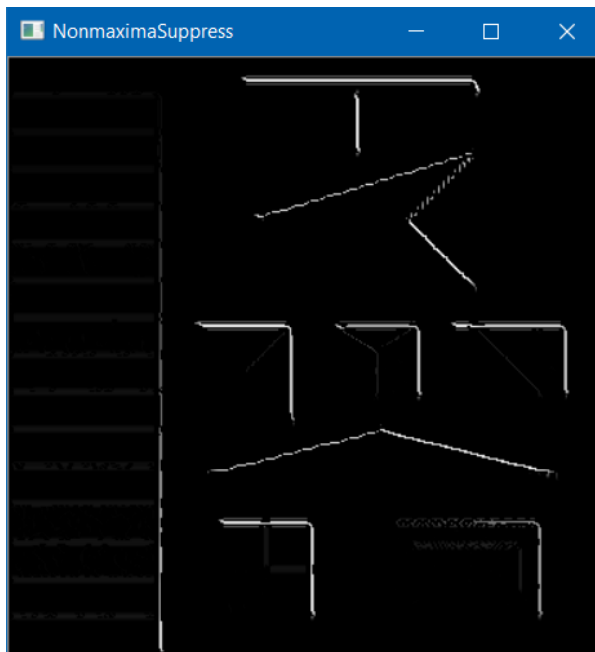
Gaussian Smoothed



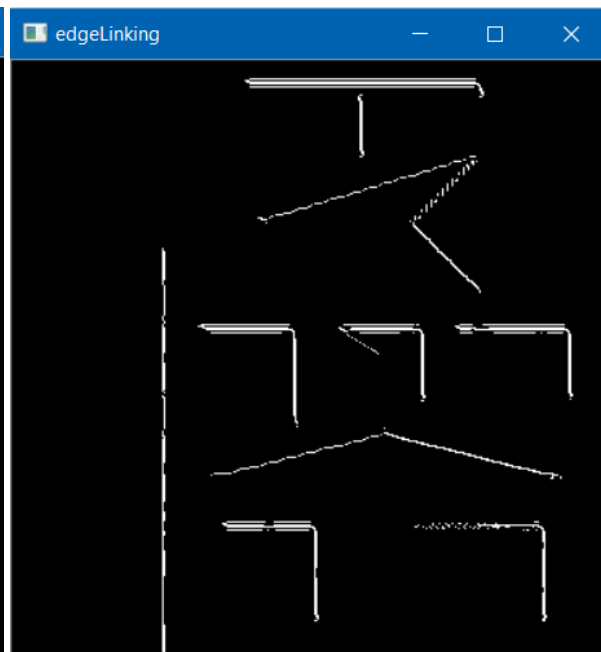
Gradient Image



Suppressed Non maxima image

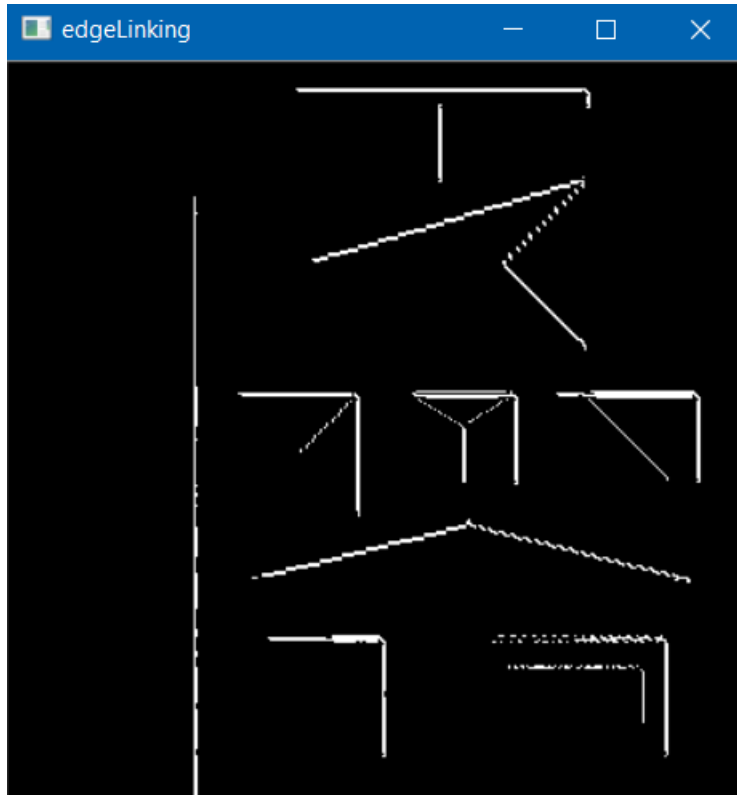


Edge Linking Image



Based on our observations: My optimal parameter choice is:

For Test image (Kernel = (3,1.5) ; LowThresh, HighThresh= 0.02,0.08)



Observations:

Similarly to the Lena image.

Kernels: Compared to 5 kernel, 3 kernel gives a crisper edge link images. I think it is because the 5 kernel is catching information from the different shapes and affecting the other shapes. 3 kernel is not able to reach the different shapes.

We are losing a lot of info in the final result with HighThresh>0.1 and catching too much info for LowThresh <0.02

Therefore, our choice of LowThresh, HighThresh= (0.02, 0.08) and kernel = (3, 1.5)