KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering

CSE 4128

Image Processing and Computer Vision Laboratory

Assignment 02

Date of Submission: 06 March, 2024

Submitted By	**Submitted To**
Md Abu Saeed Roll: 1907057 Year: 4th Semester: 1st Department of Computer Science and Engineering (CSE) Khulna University of Engineering and Technology (KUET), Khulna-9203	**Dr. Sk. Md. Masudul Ahsan** Professor **Dipannita Biswas** Lecturer Department of Computer Science and Engineering (CSE) Khulna University of Engineering and Technology (KUET), Khulna-9203

Task:

Apply **Canny edge detection** for detecting edges in images.

The steps involved in implementing the Canny edge detection algorithm:

1. **Differential operators** along x and y axis : the value of sigma will be user input.
2. **Non-maximum Suppression** finds peaks in the image gradient
3. **Hysteresis thresholding** locates edge strings

Gradient Calculation:

1. Here a Gaussian kernel is used based on user-defined sigma value.

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

2. Then the partial derivative w.r.to x and y are calculated. These are the kernel_x and kernel_y. Formulas are:

$$\frac{\partial G_\sigma(x,y)}{\partial x} = -\frac{x}{\sigma^2}G_\sigma(x,y) \qquad \frac{\partial G_\sigma(x,y)}{\partial y} = -\frac{y}{\sigma^2}G_\sigma(x,y)$$

3. The generated kernels(sigma = 0.7) are:

```
[[   1   12   54   90   54   12    1]
 [   8   98  443  731  443   98    8]
 [  18  221  993 1638  993  221   18]
 [   0    0    0    0    0    0    0]
 [ -18 -221 -993 -1638 -993 -221 -18]
 [  -8  -98 -443 -731 -443  -98   -8]
 [  -1  -12  -54  -90  -54  -12   -1]]
```

```
[[   1    8   18    0  -18   -8   -1]
 [  12   98  221    0 -221  -98  -12]
 [  54  443  993    0 -993 -443  -54]
 [  90  731 1638    0 -1638 -731 -90]
 [  54  443  993    0 -993 -443  -54]
 [  12   98  221    0 -221  -98  -12]
 [   1    8   18    0  -18   -8   -1]]
```

Fig-1: The kernel_y(left) and kernel_x(right)

4. Output Images are:

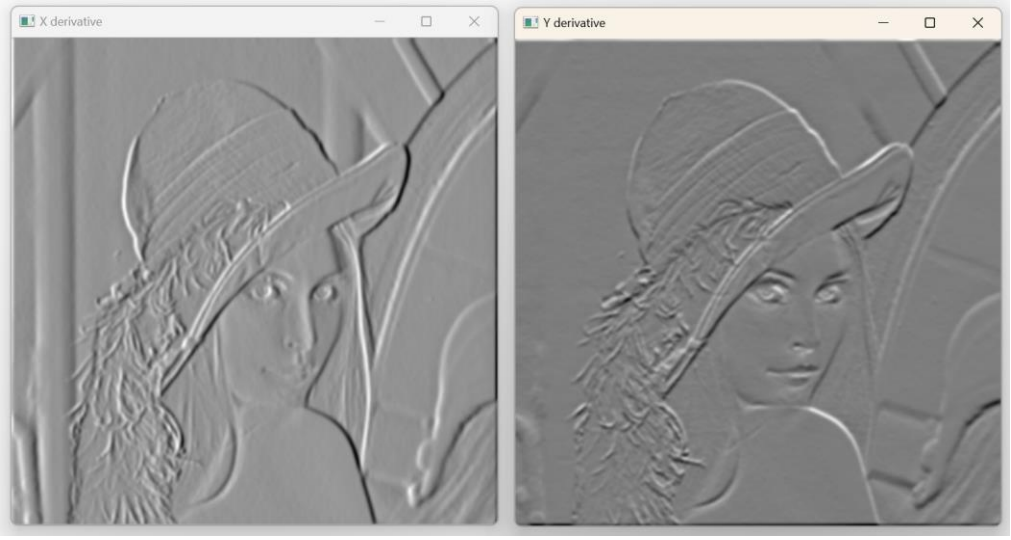

Fig-2: Convolved image using kernel_x(left) and kernel_y(right)

5. Magnitude at each pixel(x,y) is calculated using

$$\sqrt{((value\_at\_x)^2 + (value\_at\_y)^2)}$$

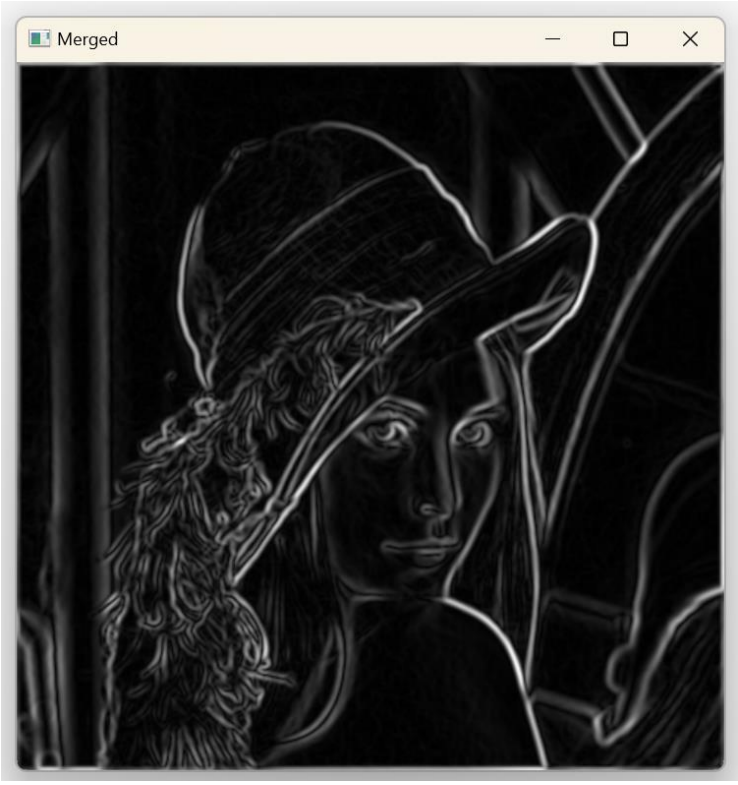

Fig-3: Magnitude of previous two images

Non-maximum Suppression:

1. It converts the detected thick edges into thin edges.
2. It uses the gradient to find the pixels to compare with.

i-1,j+1	i,j+1	i+1,j+1
i-1,j	i,j	i+1,j
i-1,j-1	i,j-1	i+1,j-1

3. If the value at comparing pixel is greater than the values of other two pixels, then the value remains same, otherwise it is made 0.
4. Sample Output

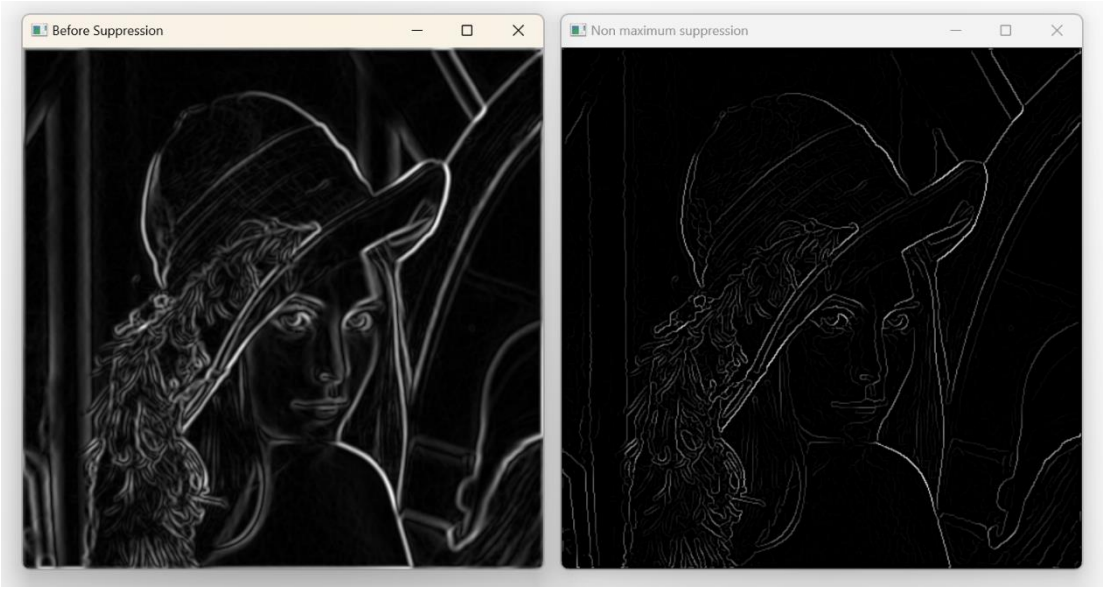

Fig-4: Result before(left) and after(right) Non-max-Suppression

Hysteresis thresholding:

1. It finds a threshold value using global thresholding technique.
2. It converts the value into lower and higher threshold value.
3. It iterates the image, and check if pixel value is
   a. larger than higher threshold, then assign 255
   b. smaller than lower threshold, then assign 0.
   c. Otherwise, assign a weak value (75).
4. After completing the dual thresholding, it performs hysteresis.

5. For each weak pixel of the image, it is connected to
   a. any strong pixel, it is made 255,
   b. otherwise, it is made 0.
6. The output is the final image.

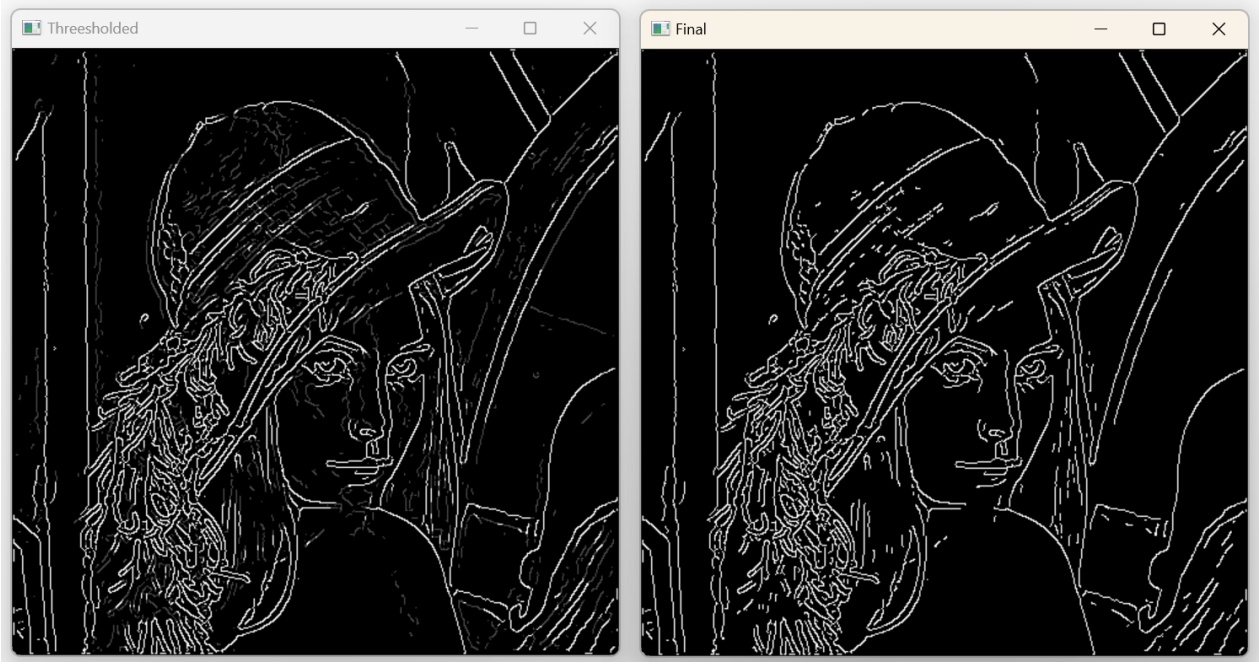

Fig-5: Image after threshold(left) and hysteresis(right)

Final Input and Output:

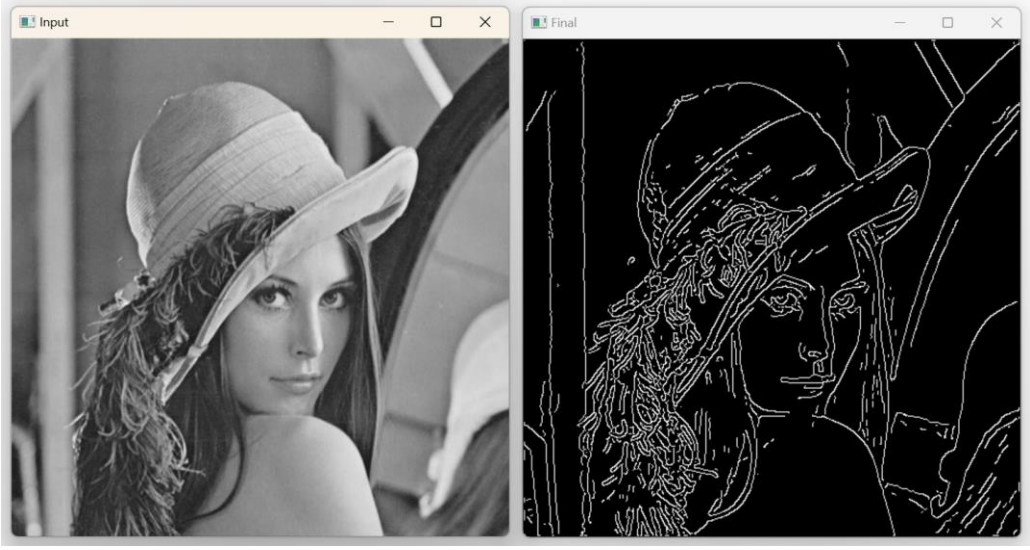

Fig-6: Final Input and Output Image