CSE573

Homework set #3

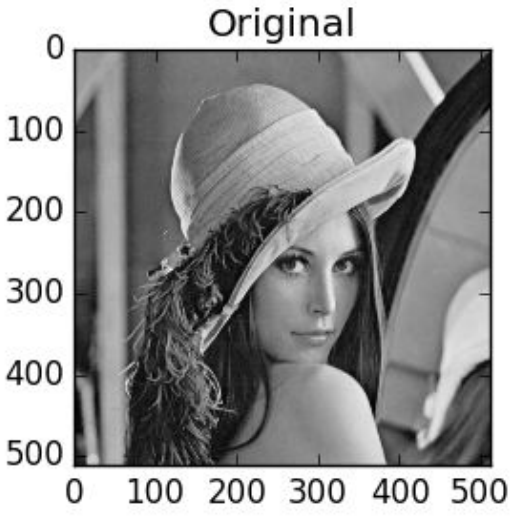
By: Priyanka Singh

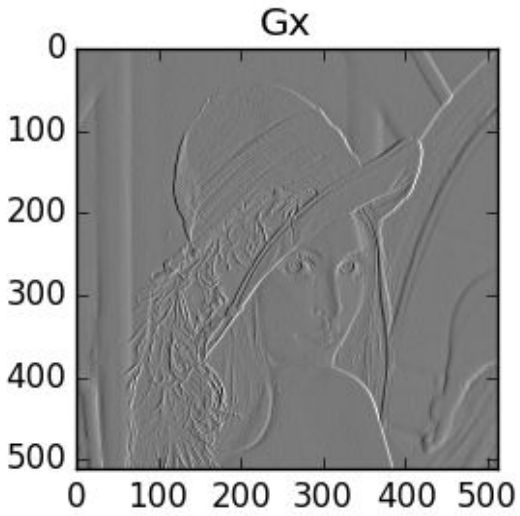
Person #: 50169994

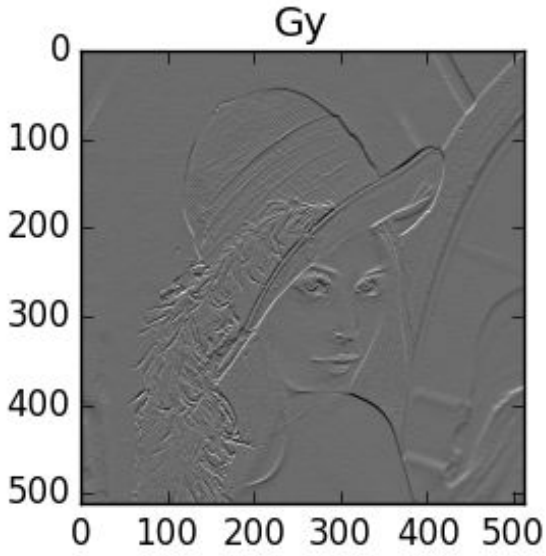
Problem 1- 1D and 2D convolution on images

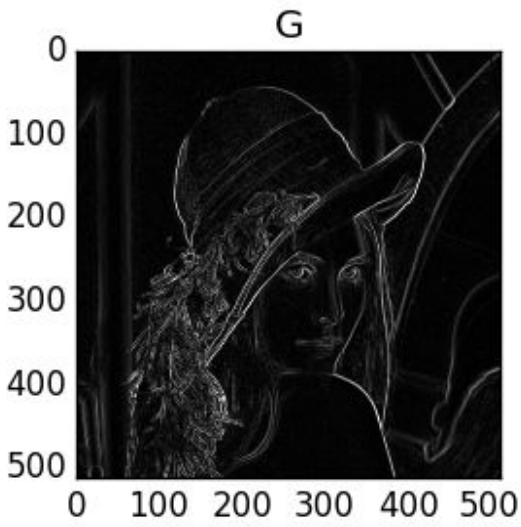
Sobel filter is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasizing edges. It computes gradient in x and y directions

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \qquad G = \sqrt{G_x^2 + G_y^2}$$





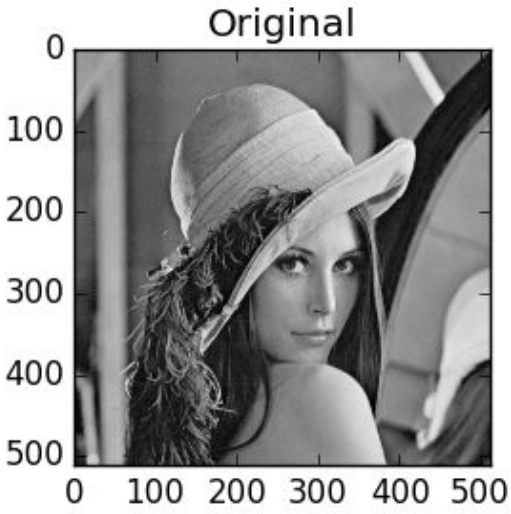




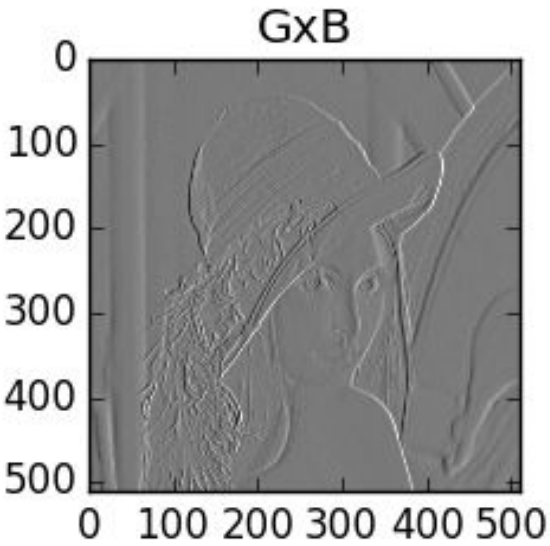
The filter kernels of Sobel filter are linearly separable

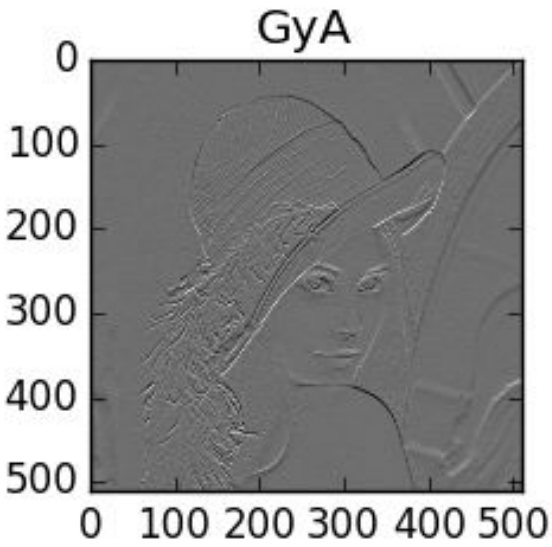
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

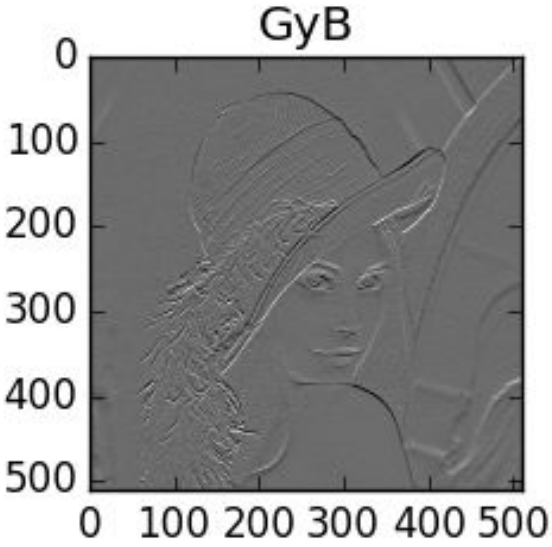
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

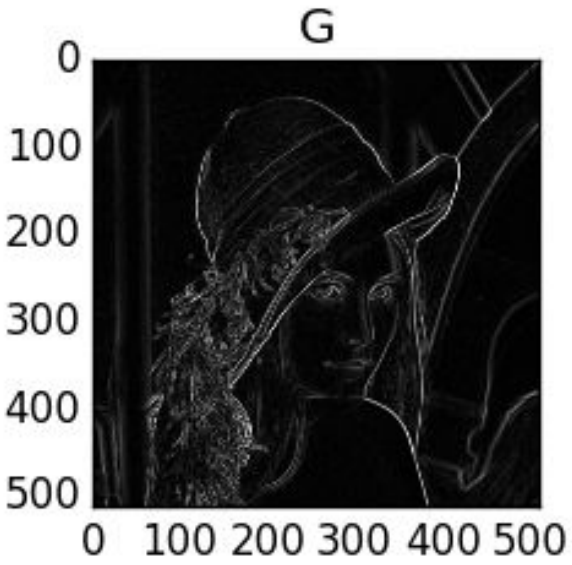



GxA
0
100
200
300
400
500
0 100 200 300 400 500


GxB
0
100
200
300
400
500
0 100 200 300 400 500


GyA
0
100
200
300
400
500
0 100 200 300 400 500


GyB
0
100
200
300
400
500
0 100 200 300 400 500



Resulting image comes out to be same in both the cases but the time taken is different.

Sobel 2D filter takes 20.8721468449 seconds to process.

Sobel 1D separable filter takes 16.1965067387 seconds to process.

Problem 2- Histogram Equilization

Histogram equalization is a technique for adjusting image intensities to enhance contrast.

Algorithm used for this process is:

1. For an $N \times M$ image of G gray-levels (often 256), create an array H of length G initialized with 0 values.
2. Form the image histogram: Scan every pixel and increment the relevant member of H—if pixel p has intensity g_p, perform

$$H[g_p] = H[g_p] + 1 .$$

3. Form the cumulative image histogram H_c:

$$H_c[0] = H[0] ,$$
$$H_c[p] = H_c[p-1] + H[p] , \quad p = 1,2,\ldots,G-1 .$$

4. Set

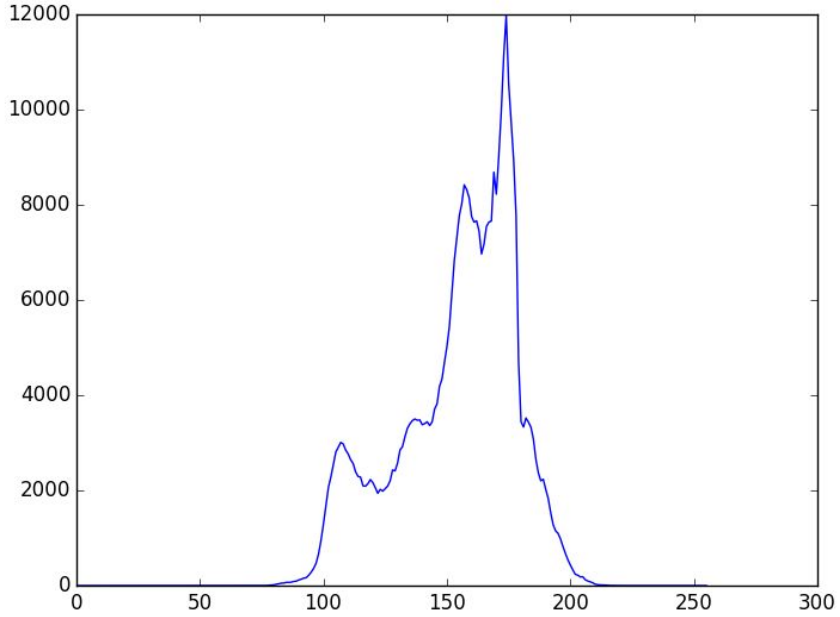
$$T[p] = \text{round}\left(\frac{G-1}{NM} H_c[p]\right) .$$

(This step obviously lends itself to more efficient implementation by constructing a look-up table of the multiples of $(G-1)/NM$, and making comparisons with the values in H_c, which are monotonically increasing.)

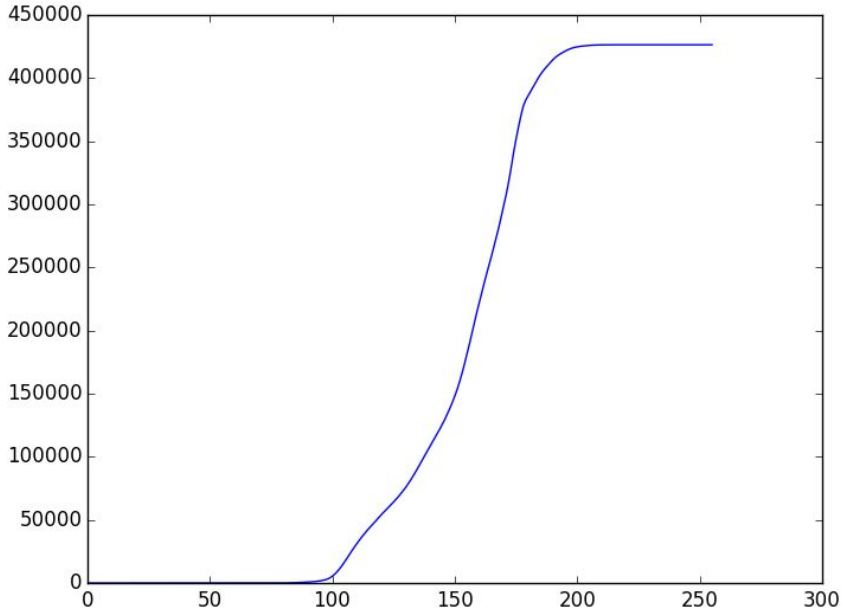
5. Rescan the image and write an output image with gray-levels g_q, setting

$$g_q = T[g_p] .$$

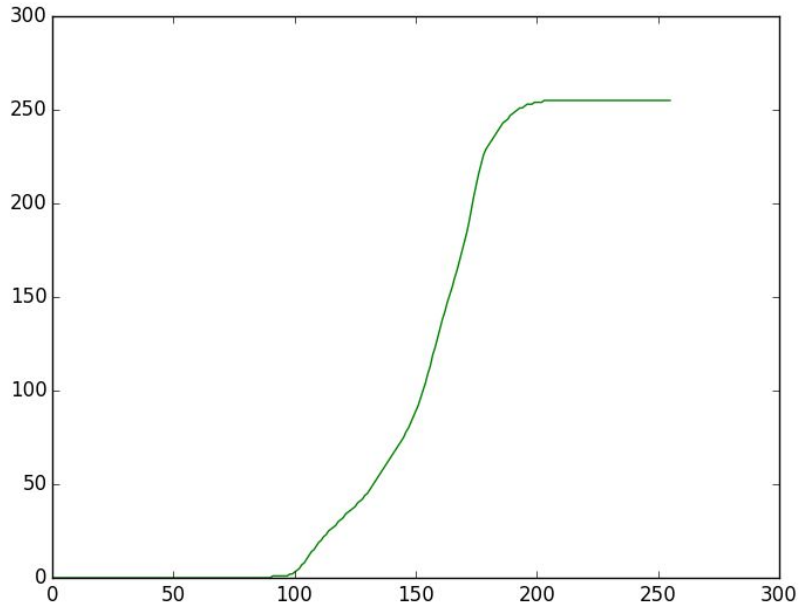
Histogram obtained for the original image in step 2:



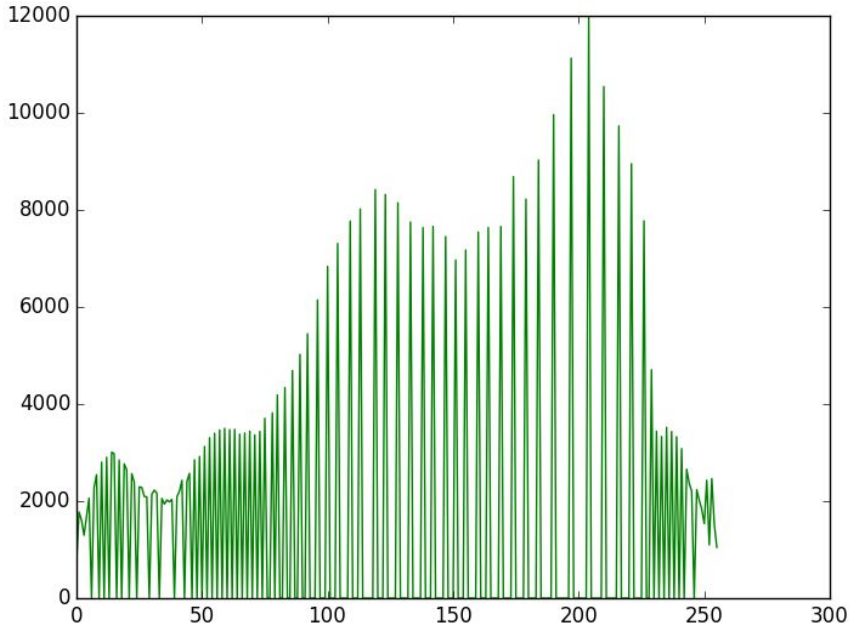
Cumulative histogram obtained in step 3:

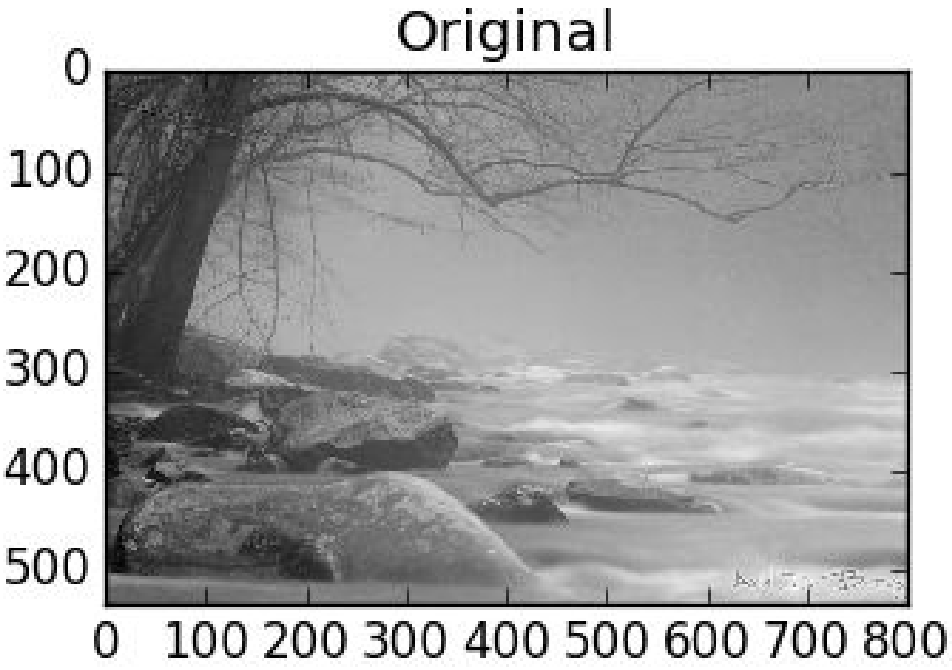


Transformation function obtained in step 4:



Histogram of the transformed image:




Original
0
100
200
300
400
500
0 100 200 300 400 500 600 700 800


new image
0
100
200
300
400
500
0 100 200 300 400 500 600 700 800

Another example