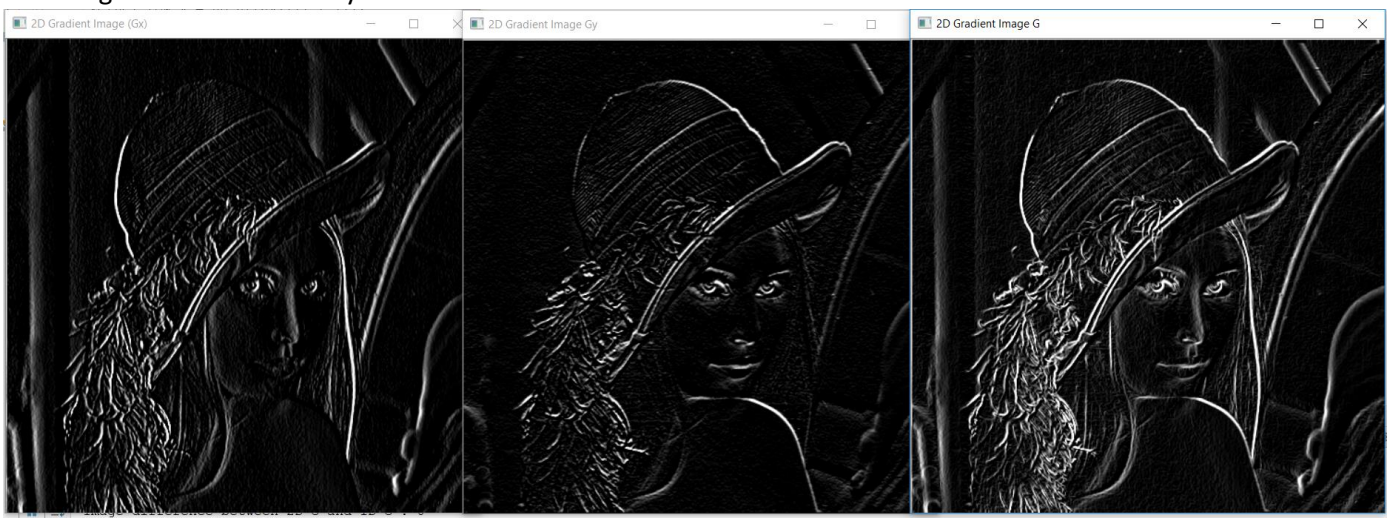
Name: Deep Narayan Mishra

Edge Detection Using Sobel Filter

1. 2D Convolution using Sobel filter –

Performed 2D convolution on image **lena_gray**.jpg with given filters (kernels) Gx, and Gy.
The image obtained for Gx Gy and G are:

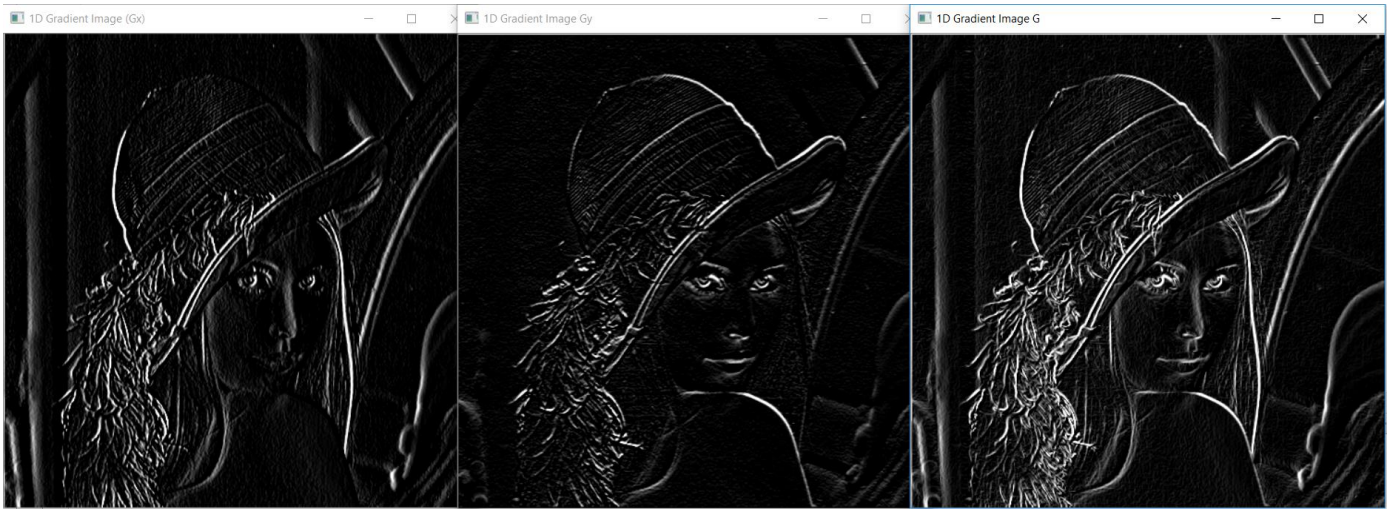


Execution time to perform 2D convolution for Gx, Gy, and G is:

```
2D Convolution using 3x3 Sobel filter
Execution time(in seconds)  5.477901494353085
```

2. 1D Convolution using Sobel filter –

Performed 1D convolution on image **lena_gray**.jpg with given filters (kernels) Gx, and Gy.
The image obtained for Gx Gy and G are:



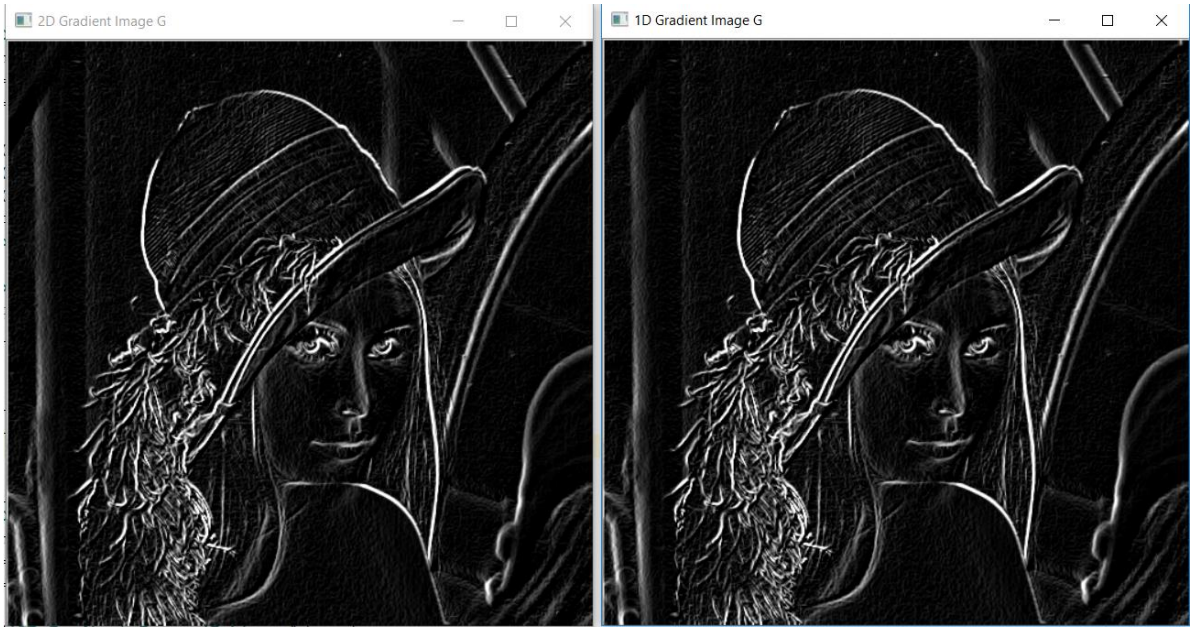
Execution time to perform 1D convolution for Gx, Gy, and G is:

```
1D Convolution using 3x3 Sobel filter
Execution time(in seconds)  6.1354445683751235
```

Apart from this, I also calculated the difference in the image matrix obtained from 1D and 2D convolution. As expected the difference is zero (means the result obtained from 1D and 2D are same):

```
Image difference between 2D-Gx-image and 1D-Gx-image : 0
Image difference between 2D-Gy-image and 1D-Gy-image : 0
Image difference between 2D-G-image and 1D-G-image : 0
```

The visual comparison of 2D and 1D convolution on image:



3. Computational complexity of 2D vs 1D:

The computational complexity of 2D convolution for MxN image and PxQ filter would be: MxN per pixel and MxNxPxQ for entire image. The reason is that for a filter of 3x3 requires 9 multiplications to convolute with 3x3 image to generate 1 pixel. Similarly, to convolute a PxQ pixels of image with PxQ filter requires PxQ multiplication to create 1 pixel. Hence PxQxMxN to generate (convolute) the entire image.

The computational complexity of 1D convolution for MxN image and separable filters (PxQ dimension of combined) would be: M+N per pixel and (M+N)xPxQ for entire image. The reason is that in separable filter, we would two kernel one with Px1 dimension and other with 1xQ dimension. The Px1 requires P multiplication, and 1xQ requires Q multiplication two convolute the image. Total P+Q multiplication requires for each pixel and (P+Q)xMxN to generate whole image.

However, due to addition buffering requirement of temporary data (convoluted only with one separable filter yet), the computation may not give as much time difference as expected sometimes (when ran on 3x3 filter). In the next section I am showing the time difference obtained between 1D and 2D convolution for 100x100 filter.

4. 2D and 1D convolution using 100x100 Sobel filter:

To perform this task: I create two separable filters, one with 100x1 dimension and another with 1x100 dimension. I created another filter of 100x100 dimension by multiplying the two separable filters.

I passed them to convolute given image, once using 100x100 filter (2D Convolution) and next time using separable filters (1D Convolution).

I captured the time difference to execute these convolution: The difference is very significant

```
2D Convolution using 100x100 Sobel filter
Execution time(in seconds)  7.742880887347013
1D Convolution using 100x100 sobel filter
Execution time(in seconds)  3.630244514305435
Image difference between 2D-Gx-image and 1D-Gx-image for 100x100 filter : 0
```