

## DIS Project 2

Shahil Manoj Dhotre (sdhotre)

Problem 1: 2D convolution:

1a) Convolution using several padding and kernels

Type of paddings:

- Zero
- Wrap Around
- Copy Edge
- Reflect Across Edge

Type of kernels:

- Box Filter
- The simple first-order derivative filter (row and column)
- Prewitt M<sub>x</sub>
- Prewitt M<sub>y</sub>
- Sobel M<sub>x</sub>
- Sobel M<sub>y</sub>
- Roberts M<sub>x</sub>
- Roberts M<sub>y</sub>

Output:

Gray image padding output:



Zero padding



wrap-around padding



Copy edges padding



Reflect across padding

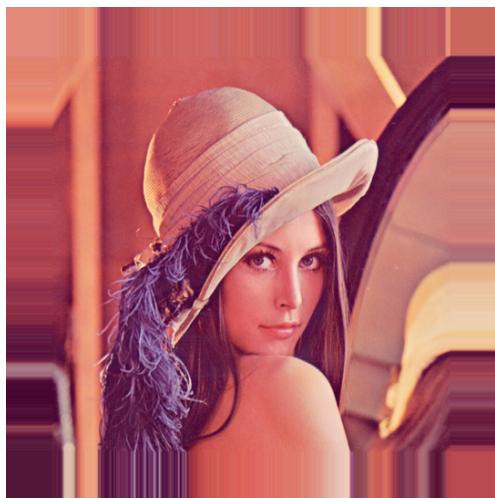
#### Color Image padding:



Zero padding



wrap-around padding



Copy edge padding

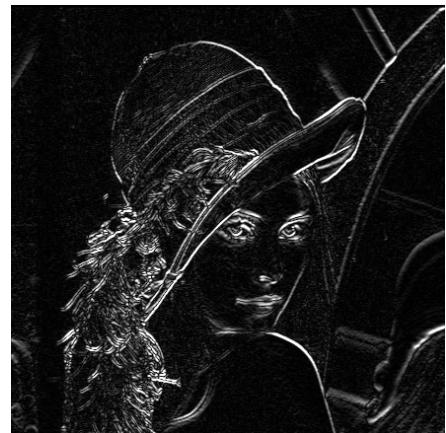


reflect across padding

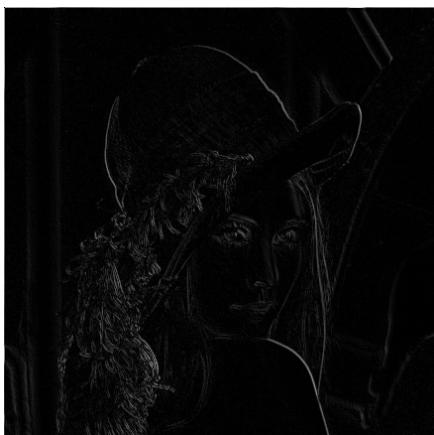
Convolution output for gray image: (for all kernels with zero padding and stride 1)



Sobel mx



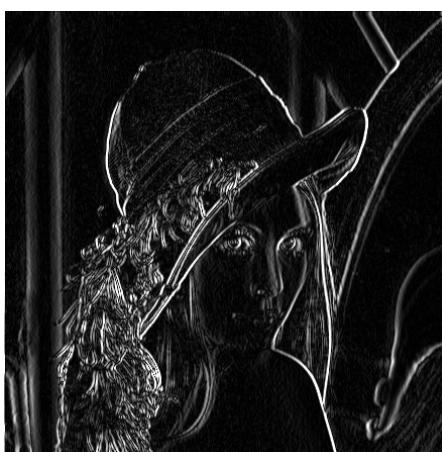
Sobel my



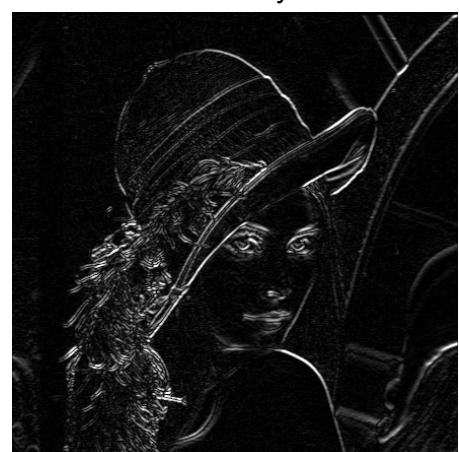
Robert mx



Robert my



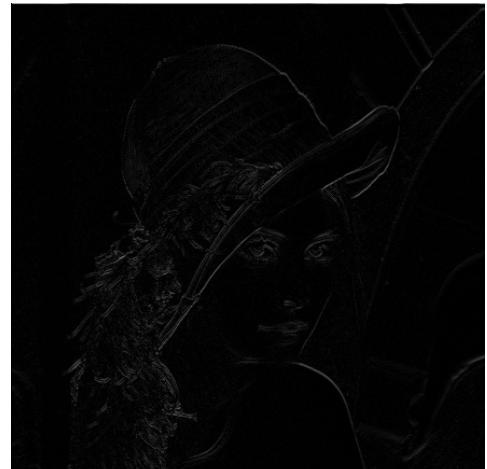
Prewitt mx



Prewitt my



First order derivative row

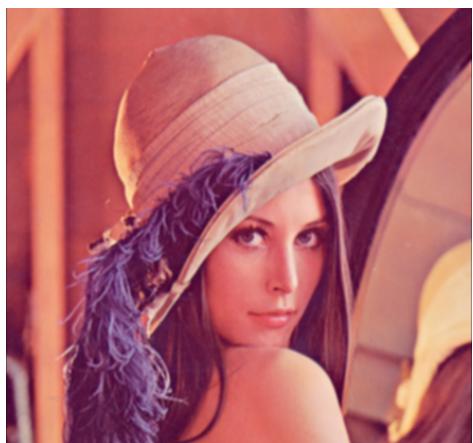


First order derivative column



Box Filter

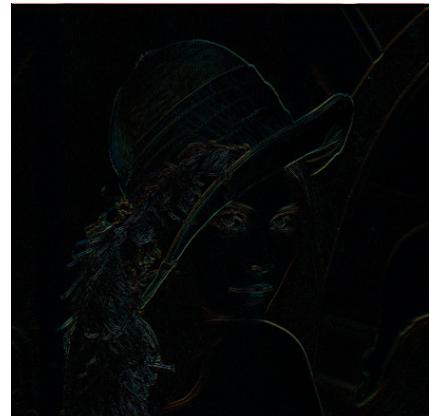
Convolution output for color image: (for all kernels and zero padding and stride 1)



Box Filter



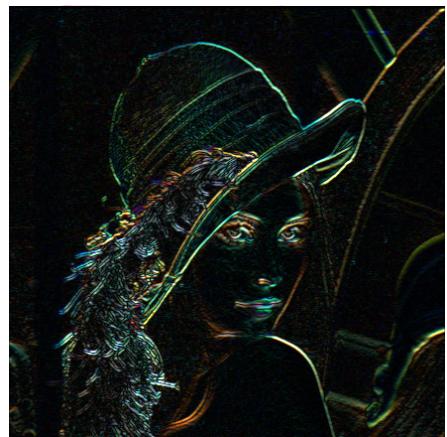
First order derivative filter row



First order derivative filter column



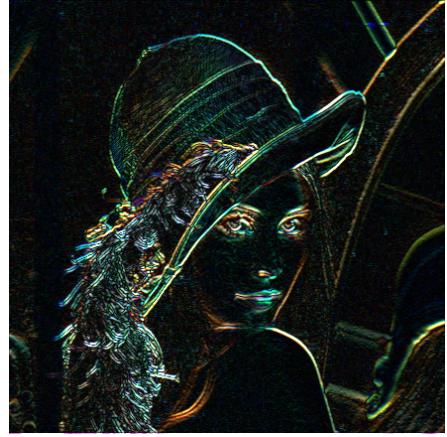
Prewitt mx



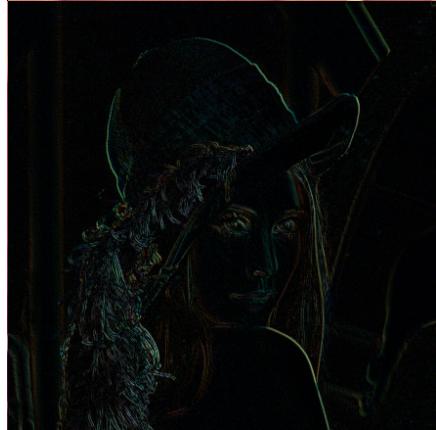
Prewitt my



Sobel mx



Sobel my



Roberts mx



Robert my

Please find more output images in a folder named as

- 1)sdhotre\_lena\_images\_zero (All kernel, zero padding)
- 2)sdhotre\_lena\_images\_wrap (All kernel, wrap-around padding)
- 3)sdhotre\_lena\_images\_copy (All kernel, copy edge padding)
- 4)sdhotre\_lena\_images\_reflect (All kernel, reflect across padding)
- 5)sdhotre\_wolf\_images\_zero (All kernel, zero padding)
- 6)sdhotre\_wolf\_images\_wrap (All kernel, wrap-around padding)
- 7)sdhotre\_wolf\_images\_copy (All kernel, copy edge padding)
- 8)sdhotre\_wolf\_images\_reflect (All kernel, reflect across padding)

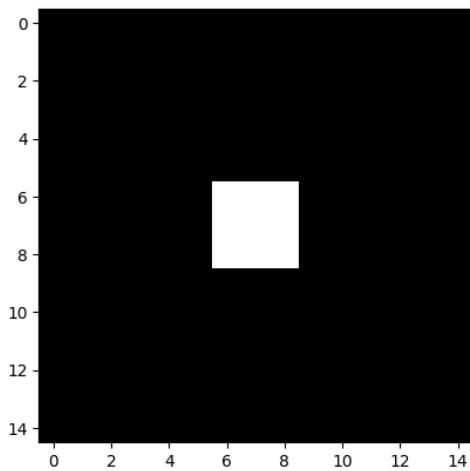
Q1b) gray impulse image of size 1024x1024 having value 1 at position (512, 512) and zero elsewhere

Output:

Convolution of gray impulse image using box kernel and zero padding



Zoomed image of the output:



The position where the values changed:

Position	convoluted_values
(511, 511)	0.1111111111111111
(511, 512)	0.1111111111111111
(511, 513)	0.1111111111111111
(512, 511)	0.1111111111111111
(512, 512)	0.1111111111111111
(512, 513)	0.1111111111111111
(513, 511)	0.1111111111111111
(513, 512)	0.1111111111111111
(513, 513)	0.1111111111111111

Please find the code for problem no. 1a and 1b in sdhotre\_project02\_Q1.ipynb

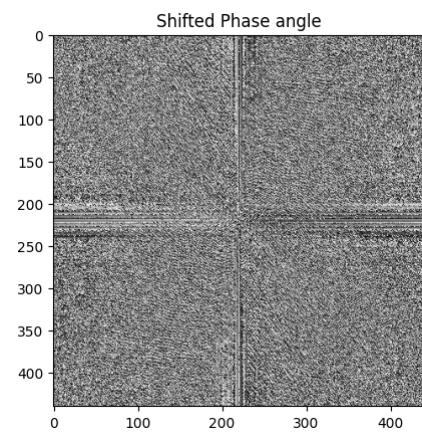
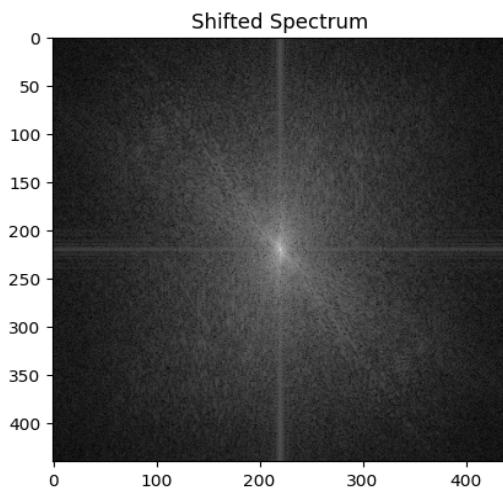
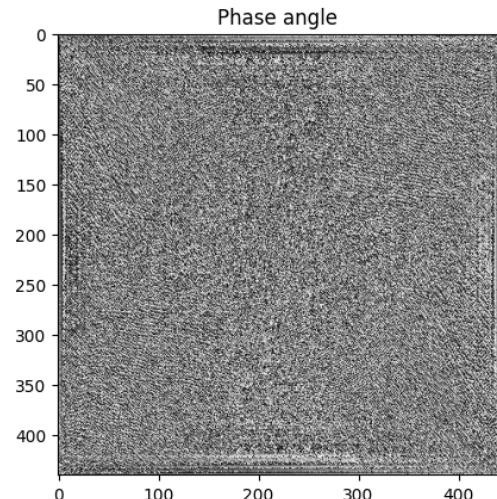
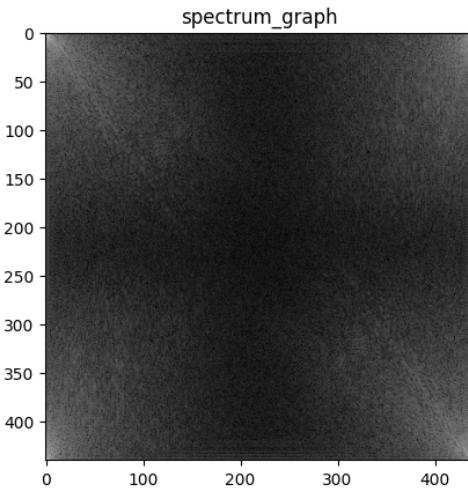
## Q2Fast Fourier Transform 2

The 2D FFT is calculated by using 1D FFT on each row and then the 1D FFT of each column of the array

a) DFT2 Lena grayscale image



Lena



```
In [240]: fft2 = fast_fourier_transform_2(gray01)
fft2

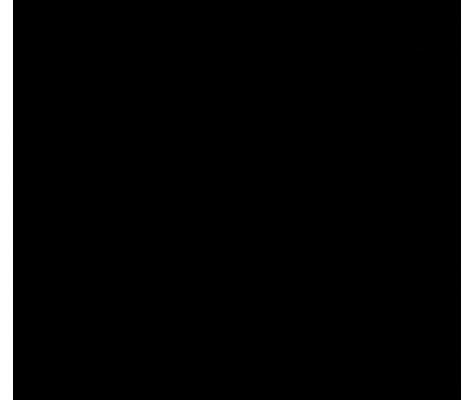
Out[240]: array([[ 9.41778902e+04 -0.j, -8.41189078e+01-3545.80433177j,
   -1.30239332e+03 -475.18786214j, ...,
   1.49444445e+03 -627.5183275j, -1.30239332e+03 +475.18786214j,
   -8.41189078e+01+3545.80433177j, ...,
  -1.32596224e+03+6928.27330717j, -4.60579307e+03+4606.0704462j ,
  -7.95677284e+02 -387.06775305j, ...,
  2.13388469e+03+2046.88293665j, 2.37846421e+03 +182.83995109j,
  1.02163145e+02-2741.29641947j, ...,
  4.45440020e+03-3279.87732399j, -3.20385249e+03 +217.4324016j ,
  1.93761483e+03-1312.06388217j, ...,
  -1.52543849e+03 -580.03115988j, -2.09541766e+03 +509.00010174j,
  -1.02766901e+03-1296.60770959j, ...,
  -6.91382986e+02-1838.43969325j, 1.89209238e+03+2351.75477524j,
  1.02386962e+03+2684.41831225j, ...,
  2.87897027e+03 +316.62592084j, -1.50092006e+03-2181.39611196j,
  -1.48180085e+03+5509.28360041j, ...,
  4.45440020e+03+3279.87732399j, -1.02766901e+03+1296.60770959j,
  -2.09541766e+03 -509.00010174j, ...,
  3.76686073e+02-1965.90569523j, 1.93761483e+03+1312.06388217j,
  -3.20385249e+03 -217.4324016j , ...,
  -1.32596224e+03-6928.27330717j, 1.02163145e+02+2741.29641947j,
  2.37846421e+03 -182.83995109j, ...,
  -1.55947281e+02+1749.06732331j, -7.95677284e+02 +387.06775305j,
  -4.60579307e+03-4606.0704462j ]])
```

DFT2 of a grayscale image

Q2b) IDFT2 is calculated using the butterfly technique that is by taking the conjugate of fft2 and then applying the 2D fft algorithm and again taking the conjugate.



Idft2 of 2D FFT image



d= f-g

d values are

---

```
In [239]: d
```

```
Out[239]: array([[1, 1, 1, ..., 0, 1, 1],  
                  [1, 1, 1, ..., 1, 1, 1],  
                  [1, 1, 1, ..., 1, 1, 0],  
                  ...,  
                  [0, 1, 0, ..., 1, 1, 1],  
                  [0, 1, 1, ..., 0, 1, 0],  
                  [0, 1, 0, ..., 0, 0, 1]], dtype=uint8)
```

---

For code run on lena image please see the sdhotre\_project2\_Q2\_lena.ipynb

For code run on the wolf image please see the sdhotre\_project2\_Q2\_wolf.ipynb