teamG_Khadga_Shashank

January 27, 2022

0.1 Team : G

0.1.1 Alli Khadga Jyoth - 19024

0.1.2 Shashank Singh - 19289

Q3

```
[1]: import numpy as np
     import cv2
```

```
[5]: M = np.array([[0.7679,-0.4938,-0.0234,0.0067],[-0.0852,-0.0915,-0.9065,-0.
      ↪0878],[0.1827,0.2988,-0.0742,1.000]])
     M
```

```
[5]: array([[ 0.7679, -0.4938, -0.0234,  0.0067],
            [-0.0852, -0.0915, -0.9065, -0.0878],
            [ 0.1827,  0.2988, -0.0742,  1.    ]])
```

```
[6]: H = M[:,:3]
     h = M[:,3].reshape(3,1)
     H_inv = np.linalg.inv(H)
     H_inv
```

```
[6]: array([[ 0.93072703, -0.14626022,  1.49334069],
            [-0.57636511, -0.1766679 ,  2.3401131 ],
            [-0.02930009, -1.07156483, -0.37656147]])
```

```
[8]: R_T,K_inv = np.linalg.qr(H_inv)
     R_T
```

```
[8]: array([[-0.84987872, -0.13141145, -0.51033048],
            [ 0.52629872, -0.16249257, -0.83462915],
            [ 0.02675492, -0.97791983,  0.20726065]])
```

```
[9]: K_inv
```

```
[9]: array([[-1.09512923e+00,  2.65373217e-03, -4.76348136e-02],
            [ 0.00000000e+00,  1.09583198e+00, -2.08246122e-01],
            [ 0.00000000e+00,  0.00000000e+00, -2.79327026e+00]])
```

```
[11]: t = K_inv@h
      t
```

```
[11]: array([[-0.05520518],
             [-0.30446017],
             [-2.79327026]])
```

0.1.3 Q18

```
[20]: img = cv2.imread('noisy-grayscale.png')
      # img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY).astype(float)
      # img
```

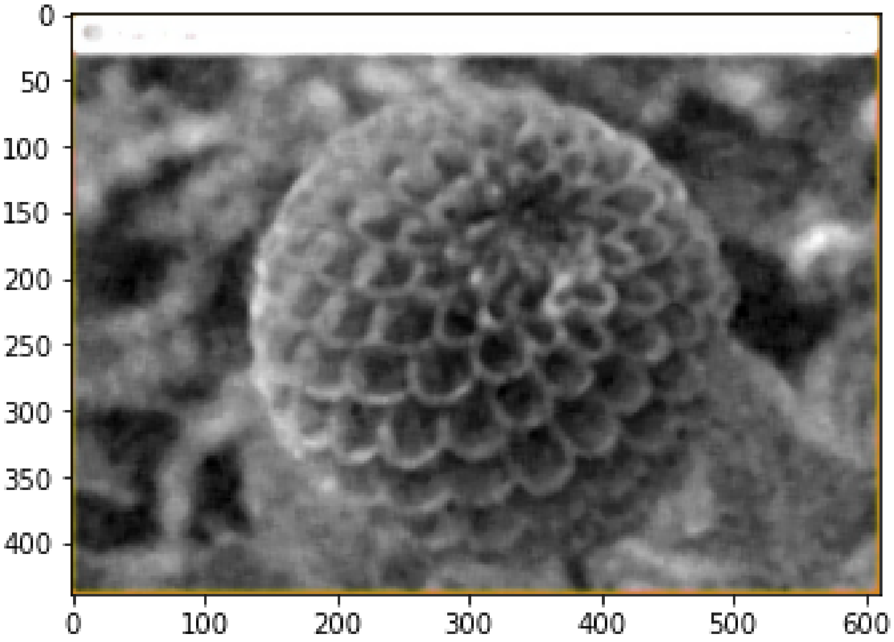
Denoising the image using median blur as the image has salt-pepper noise

```
[26]: imgf = cv2.medianBlur(img,9)
      cv2.imshow('Blurred Image', imgf)
      cv2.imwrite('denoise.jpg',imgf)
      # cv2.waitKey(0)
      # cv2.destroyAllWindows()
```

```
[26]: True
```

```
[33]: import matplotlib.pyplot as plt
      img_de  = cv2.imread('denoise.jpg')
      plt.imshow(img_de)
```

```
[33]: <matplotlib.image.AxesImage at 0x2987a79c970>
```

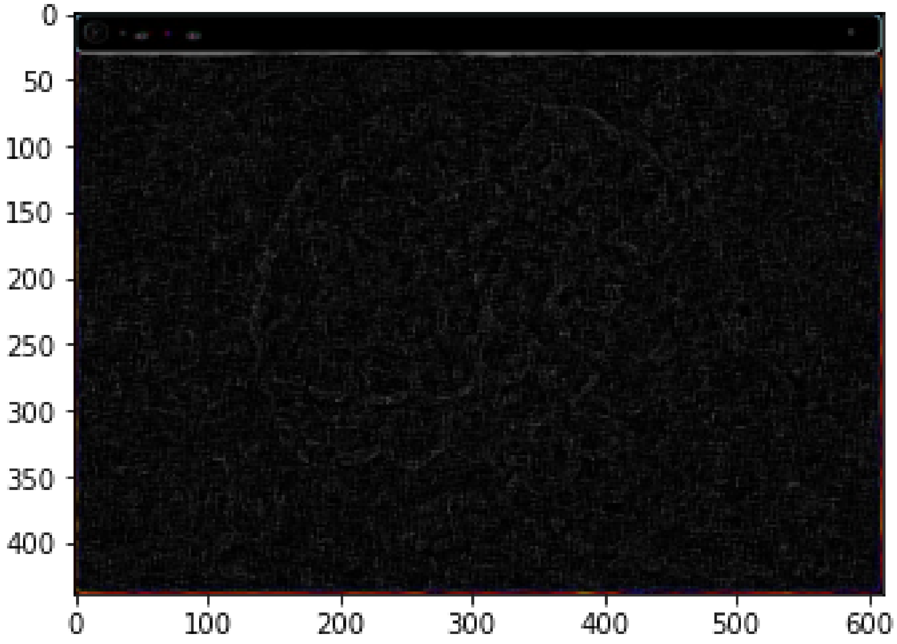


[34]:
```
kernel = np.array([[-1.0, -1.0, -1.0],
                   [-1.0, 8.0, -1.0],
                   [-1.0, -1.0, -1.0]])

kernel = kernel/(np.sum(kernel) if np.sum(kernel)!=0 else 1)

#filter the source image
img_hpf = cv2.filter2D(imgf,-1,kernel)

#save result image
cv2.imwrite('denoice_filtered.jpg',img_hpf)
```

[34]:
```
True
```

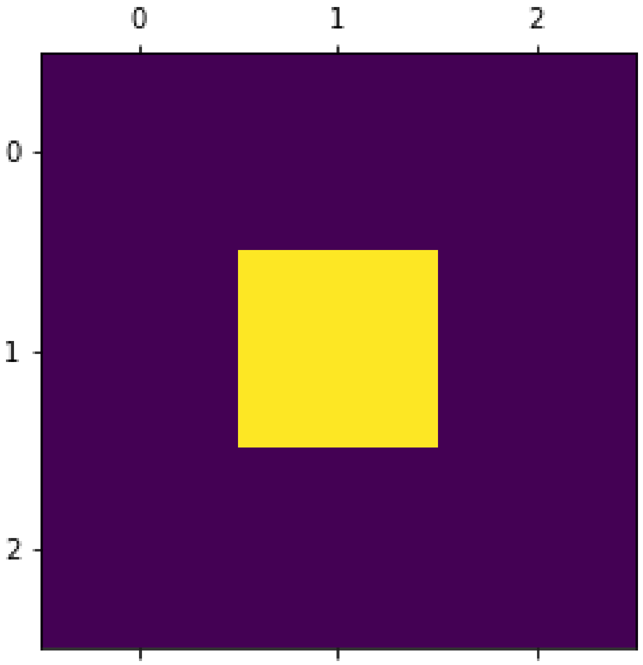
[35]:
```
img_hp = cv2.imread('denoice_filtered.jpg')
plt.imshow(img_hp)
```

[35]:
```
<matplotlib.image.AxesImage at 0x2987b7cb430>
```



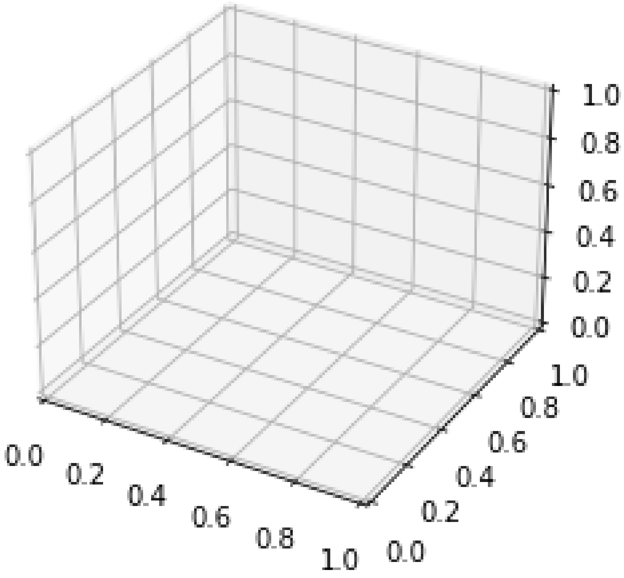
[37]:
```
plt.matshow(kernel)
```

[37]:
```
<matplotlib.image.AxesImage at 0x2987b7f5070>
```



[39]:
```
plt.rcParams["figure.figsize"] = [7.00, 3.50]
plt.rcParams["figure.autolayout"] = True
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
data = kernel
z, x, y = data.nonzero()
ax.scatter(x, y, z, c=z, alpha=1)
plt.show()
```

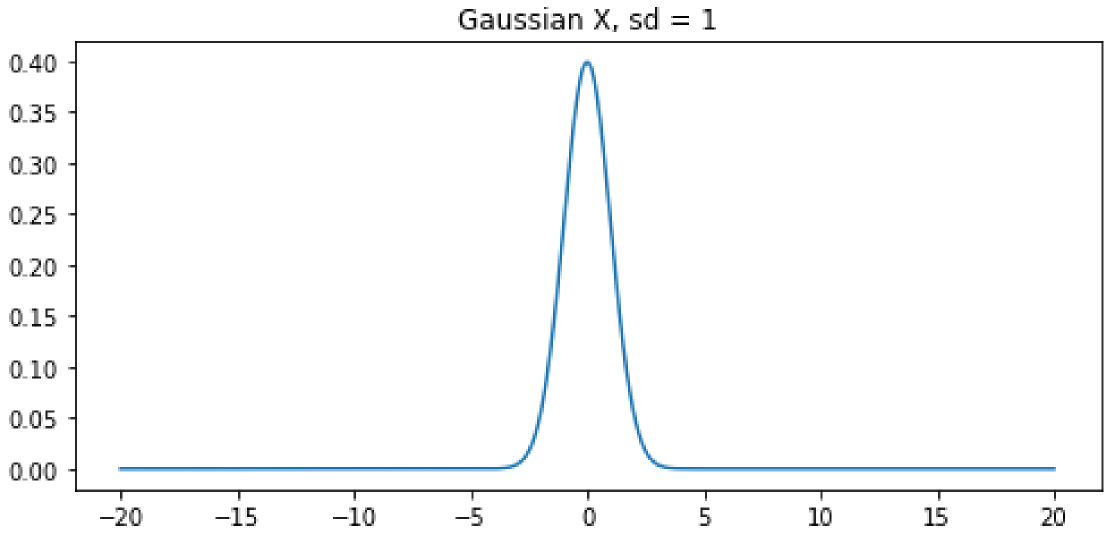
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\KHADGA~1\AppData\Local\Temp/ipykernel_20216/3174348610.py in <module>
      4 ax = fig.add_subplot(111, projection='3d')
      5 data = kernel
----> 6 z, x, y = data.nonzero()
      7 ax.scatter(x, y, z, c=z, alpha=1)
      8 plt.show()

ValueError: not enough values to unpack (expected 3, got 2)
```



0.1.4 Q 15

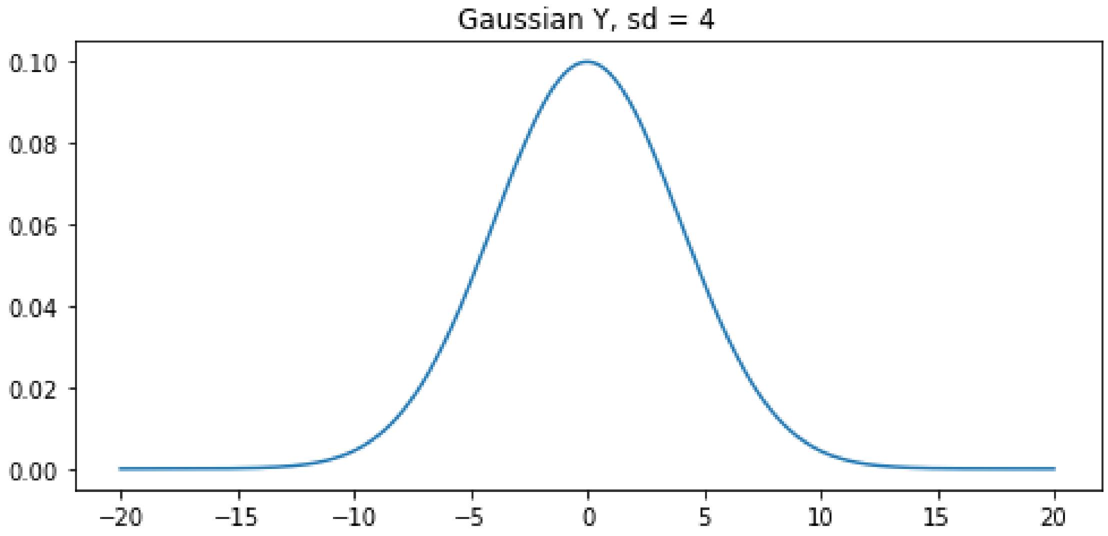
[47]:

```
from scipy.stats import norm
import statistics
# Plot between -10 and 10 with .001 steps.
x_axis = np.arange(-20, 20, 0.01)

# Calculating mean and standard deviation
mean = statistics.mean(x_axis)
sd = 1

plt.plot(x_axis, norm.pdf(x_axis, mean, sd))
plt.title('Gaussian X, sd = 1')
plt.show()
```



[48]:

```
# Plot between -10 and 10 with .001 steps.
x_axis = np.arange(-20, 20, 0.01)

# Calculating mean and standard deviation
mean = statistics.mean(x_axis)
# sd = statistics.stdev(x_axis)
sd = 4

plt.plot(x_axis, norm.pdf(x_axis, mean, sd))
plt.title('Gaussian Y, sd = 4')
plt.show()
```



0.1.5 Q14

After blurring with box kernal of size (3,3) the first image is only blurred at the edge of gray and blackand the second image is blurred throughout since the boxes are small. So, the second image is more affected than the frst image. SO the histograms of the images will be different.

0.1.6 Q9

Rough work

```
[50]: R = np.array([[0.9,0.4,0.1732],[-0.4183,0.9043,0.0854],[-0.1225,-0.1493,0.
       ↪9812]])
      Cw = np.array([[-1],[-2],[-3]])
```

```
[50]: array([[2.2196],
             [1.6465],
             [2.5225]])
```

```
[51]: R
```

```
[51]: array([[ 0.9   ,  0.4   ,  0.1732],
             [-0.4183,  0.9043,  0.0854],
             [-0.1225, -0.1493,  0.9812]])
```

```
[52]: Cw
```

```
[52]: array([[-1],
             [-2],
             [-3]])
```

```
[53]: -R@Cw
```

```
[53]: array([[2.2196],
             [1.6465],
             [2.5225]])
```

```
[ ]:
```