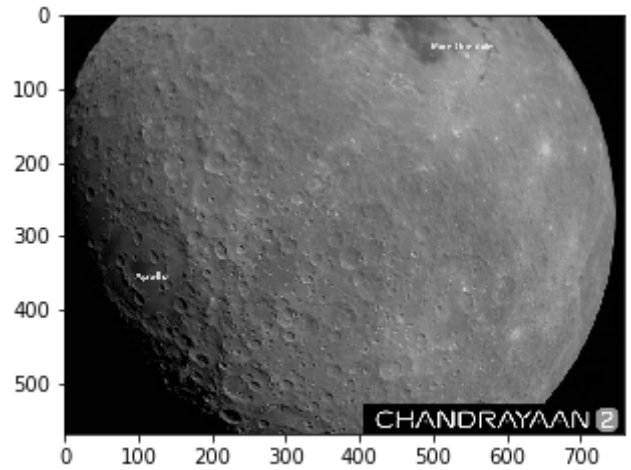
Question 3 a

```
import matplotlib.pyplot as plt
import numpy as np
import string
import cv2
import random
from scipy import ndimage
import math
```

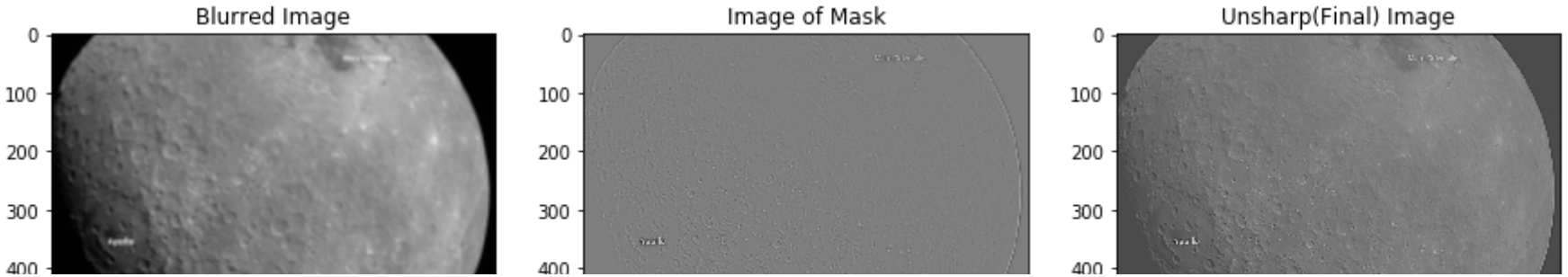
```
img = cv2.imread('Chandrayaan2 - Q3a-inputimage.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray=gray.astype(np.float)

imgg = cv2.imread('Q3-a unsharpmasked output.jpg')
grayy = cv2.cvtColor(imgg, cv2.COLOR_BGR2GRAY)
# grayy=grayy.astype(np.float)
```

```
plt.figure(figsize=(5, 5))
plt.imshow(gray,cmap='gray')
plt.show()
```

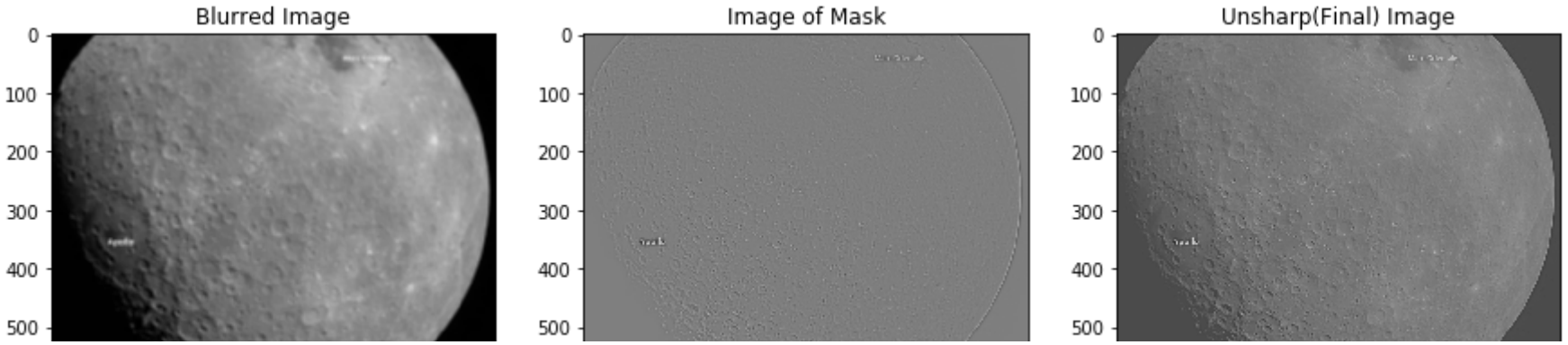


```
a=int((7-1)/2)
b=int((7-1)/2)
filter=(1/49)*np.ones((7,7))
```

```
blur2=ndimage.convolve(gray,filter)
fig, axs = plt.subplots(1, 3, figsize=(15, 15))
axs[0].set_title("Blurred Image")
axs[0].imshow(blur2,cmap='gray')


mask2=gray-blur2
axs[1].set_title("Image of Mask")
axs[1].imshow(mask2,cmap='gray')


unsharpmask2=gray+mask2
# unsharpmask2=cv2.normalize(unsharpmask2,None,0,255,cv2.NORM_MINMAX)

cv2.imwrite("unsharpmask2.png",unsharpmask2)
axs[2].set_title("Unsharp(Final) Image")
axs[2].imshow(np.round(unsharpmask2),cmap='gray')

plt.show()
```



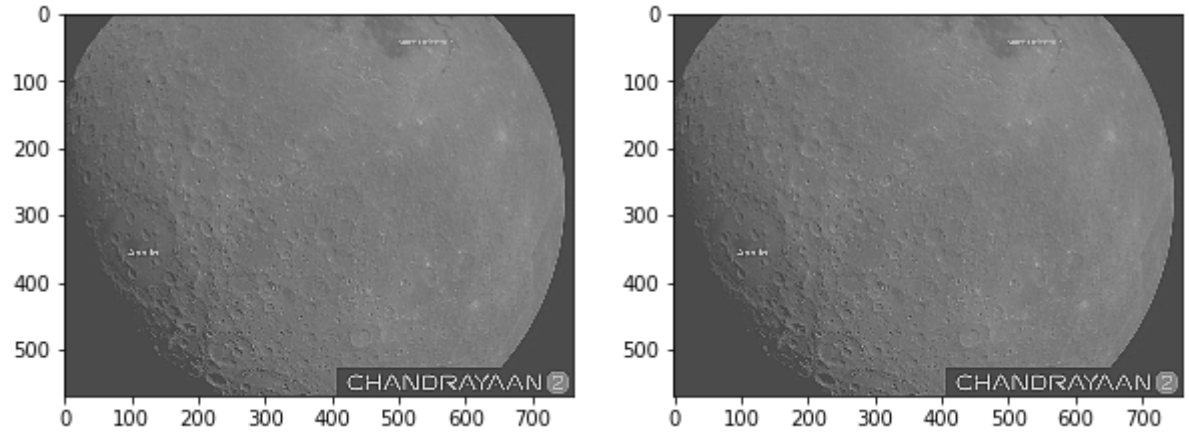

```
blur=np.zeros((gray.shape))

for i in range(0,gray.shape[0]):
    for j in range(0,gray.shape[1]):
        val=0
        for k in range(-1*a,a+1):
            for l in range(-1*b,b+1):
                if(i-k>=0 and j-l>=0 and i-k<gray.shape[0] and j-l <gray.shape[1]):
                    val+=filter[k+a][l+b]*gray[i-k][j-l]
        blur[i][j]=val
fig, axs = plt.subplots(1, 3, figsize=(15, 15))
axs[0].set_title("Blurred Image")
axs[0].imshow(blur,cmap='gray')


mask=gray-blur
axs[1].set_title("Image of Mask")
axs[1].imshow(mask,cmap='gray')


unsharpmask=gray+mask
unse=np.zeros((unsharpmask.shape[0],unsharpmask.shape[1]))
for i in range(0,unsharpmask.shape[0]):
    for j in range(0,unsharpmask.shape[1]):
        if unsharpmask[i][j]>=0:
            unse[i][j]=unsharpmask[i][j]

# unsharpmask=cv2.normalize(unsharpmask,None,0,255,cv2.NORM_MINMAX)
cv2.imwrite("unsharpmask.png",unsharpmask)
axs[2].set_title("Unsharp(Final) Image")
axs[2].imshow(np.round(unsharpmask),cmap='gray')

plt.show()
```



Question 3 c

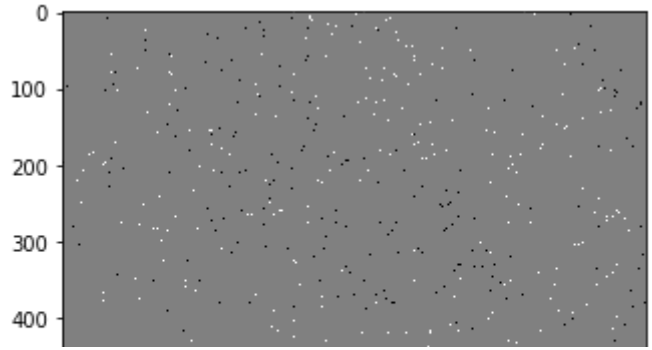
```
blur3=np.zeros((gray.shape))
# filter2=2-((1/49)*np.zeros((7,7)))
filter2=np.zeros((7,7))
filter2[3][3]=2

blur3=ndimage.convolve(gray,(filter2-filter))


blur22=np.zeros((gray.shape))
filter11=filter2-filter
for i in range(0,gray.shape[0]):
    for j in range(0,gray.shape[1]):
        val=0
        for k in range(-1*a,a+1):
            for l in range(-1*b,b+1):
                if(i-k>=0 and j-l>=0 and i-k<gray.shape[0] and j-l <gray.shape[1]):
                    val+=filter11[k+a][l+b]*gray[i-k][j-l]
        blur22[i][j]=val


# print(minun3,maxun3)
unse2=np.zeros((blur3.shape[0],blur3.shape[1]))
unse22=np.zeros((blur22.shape[0],blur22.shape[1]))
for i in range(0,blur3.shape[0]):
    for j in range(0,blur3.shape[1]):
        if blur3[i][j]>=0:
            unse2[i][j]=blur3[i][j]
        if blur22[i][j]>=0:
            unse22[i][j]=blur22[i][j]

```

```
# blur22=cv2.normalize(blur22,None,0,255,cv2.NORM_MINMAX)

cv2.imwrite("convolv.png",blur22)
fig, axs = plt.subplots(1, 2, figsize=(10, 10))
axs[0].imshow(np.round(blur3),cmap='gray')
axs[1].imshow(np.round(blur22),cmap='gray')
plt.show()
```



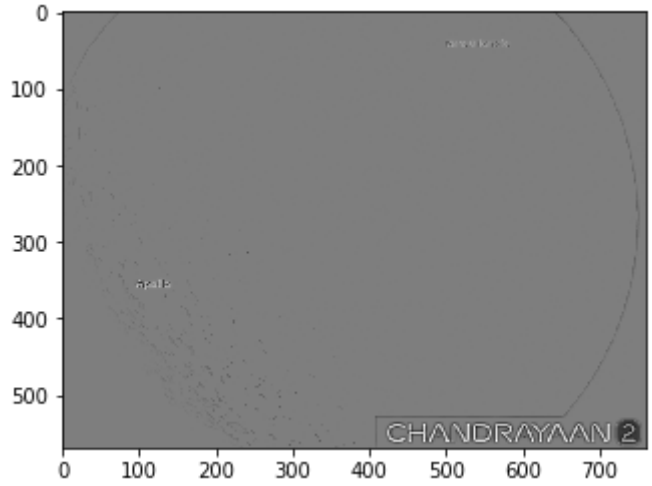

```


diffimage= np.zeros((569, 760))
for i in range(569):
    for j in range(760):
        diffimage[i][j]= int(unsharpmask[i][j]) - int(blur22[i][j])
plt.imshow(diffimage,cmap='gray')
cv2.imwrite("3a_3c.png",diffimage)
```

True



```
diffimage= np.zeros((569, 760))
tr=0
for i in range(569):
    for j in range(760):
        diffimage[i][j]= int(unsharpmask[i][j]) - int(grayy[i][j])
        if(diffimage[i][j]<=0):
            tr+=1
# print(tr)

plt.imshow(diffimage,cmap='gray')
cv2.imwrite("3a_output.png",diffimage)
```

True

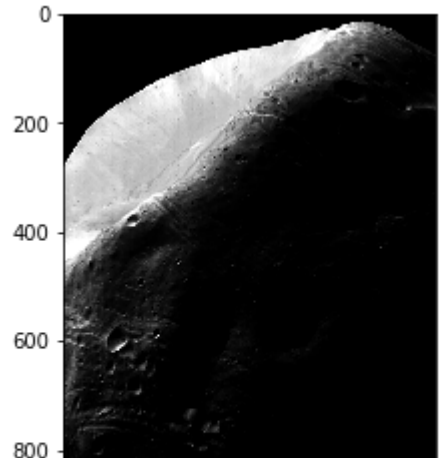


The images obtained through both the ways 3a and 3b will be very similar(identical therotically) since the convolution operator hold the distributive property. Small difference would be there due to normalization and rounding off the pixel values.

```
grau = cv2.imread('Q2-input image.tif')
image2 = cv2.cvtColor(grau, cv2.COLOR_BGR2GRAY)
```

Question 2

```
plt.figure(figsize=(5, 5))
plt.imshow(image2,cmap='gray')
plt.show()
```




```
filterr=(1/121)*np.ones((11,11))
image1= np.zeros((image2.shape[0],image2.shape[1]))

for i in range(0,image1.shape[0]):
    for j in range(0,image1.shape[1]):
        val=0
        for k in range(-1*5,5+1):
            for l in range(-1*5,5+1):
                if(i-k>=0 and j-l>=0 and i-k<image2.shape[0] and j-l <image2.shape[1]):
                    val+=filterr[k+5][l+5]*image2[i-k][j-l]
        image1[i][j]=val
```

```
plt.figure(figsize=(5, 5))
plt.imshow(image1,cmap='gray')
plt.show()
```

```
pixmaxval=int(np.amax(image1))
arr01=np.zeros((pixmaxval+1))
arr02=np.zeros((pixmaxval+1))
print(arr01.shape)
```

```
for i in range(0,image1.shape[0]):
    for j in range(0, image1.shape[1]):
        arr01[int(round(image1[i][j]))]+=1
for i in range(0,image2.shape[0]):
    for j in range(0, image2.shape[1]):
        arr02[image2[i][j]]+=1
```

```
xaxis=[]
for i in range(0,256):
    xaxis.append(i)

print(np.amax(arr02))
print(np.amax(arr01))

# plt.bar(xaxis[2:],arr02[2:])


# plt.show()

# plt.bar(xaxis[2:],arr01[2:])


# plt.show()
plt.hist(image2)
plt.show()

plt.hist(image1)
plt.show()

arr02=arr02/sum(arr02)

```

```
arr01=arr01/sum(arr01)
# arr01=np.array([1/256]*256)

print(arr01)
# plt.plot(arr01,xaxis)
```

```
ps1=np.zeros((256))
ps2=np.zeros((256))
```

```
s1=[]
for i in range(0,256):
    tempcdf=0
    for j in range(0,i+1):
        tempcdf+=arr01[j]
    tempf=(255)*tempcdf
    s1.append(tempf)

# for i in range(0,256):
#     ps1[s1[i]]=ps1[s1[i]]+arr01[i]
```

```
s2=[]
for i in range(0,256):
    tempcdf=0
    for j in range(0,i+1):
        tempcdf+=arr02[j]
    tempf=(255)*tempcdf
```

```
    s2.append(tempf)
```

```
arr03=np.zeros((256))
for i in range(0, 256):
    min= 1000000000000
    for j in s1:
        if abs(s2[i]-j)<=min:
            min=abs(s2[i]-j)
        else:
            arr03[i]=round(j)
            break
for i in range(0,255):
    print(i,arr03[i])
```

```
for i in range(0,image2.shape[0]):
    for j in range(0,image2.shape[1]):
        if(arr03[image2[i][j]]>=0):
            image2[i][j]=arr03[image2[i][j]]
        else:
            image2[i][j]=0
```

```
plt.figure(figsize=(5, 5))
plt.imshow(image2,cmap='gray')
plt.show()
```

```
plt.hist(image2)
plt.show()
```

```
# w=[[1,0,0],[0,0,0],[0,0,-1]]
# I=[[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,1,0,0,0],[0,0,0,1,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0]]
# from scipy import ndimage
# blur2=ndimage.convolve(I,w)
```

```
# print(blur2)
```

```
1
```