

4. Box filters, filter 2D, Gaussian Blur, and Median Blur, & write a program to hardcode convolution using a kernel of known size & value

```
In [2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
img=cv2.imread("black.png",0)
img1=cv2.imread("median.png")
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

```
In [3]: #box filter
img_box= cv2.boxFilter(img, -1, (9,9))
```

```
In [4]: #Gaussian blur
img_gaussian= cv2.GaussianBlur(img, (9,9),0)
```

```
In [5]: #median blur
img_medain= cv2.medianBlur(img, 5)
```

```
In [6]: #filter 2D
kernel=np.array([[ -1, -1, -1], [-1, 8, -1], [-1, -1, -1]])
img_2d= cv2.filter2D(img, -1, kernel)
```

```
In [ ]:
```

```
In [14]: def convolution(img,kernel):
w,h=img.shape[:2]
img1=img.copy()
for i in range(w-2):
    for j in range(h-2):
        mat1=np.array([img[i:i+2,j:j+2]])
        mat=np.sum(np.array(mat1*kernel))
        img1[i+1][j+1]=abs(mat)
    return img1

kernel=np.array([[ -1, -1, -1], [-1, 8, -1], [-1, -1, -1]])
k1=np.array([[1,0],[0,-1]])
conv=convolution(img,k1)
```

ValueError Traceback (most recent call last)

Cell In[14], line 13

```
11 kernel=np.array([[ -1, -1, -1], [-1, 8, -1], [-1, -1, -1]])
12 k1=np.array([[1,0],[0,-1]])
----> 13 conv=convolution(img,k1)
```

Cell In[14], line 7, in convolution(img, kernel)

```
5     for j in range(h-2):
6         mat1=np.array([img[i:i+2,j:j+2]])
----> 7         mat=np.sum(np.array(mat1*kernel))
8         img1[i+1][j+1]=abs(mat)
9     return img1
```

ValueError: operands could not be broadcast together with shapes (1,2,2,3) (2,2)

In []:

In [12]: plt.figure(figsize=(20, 15))

```
plt.subplot(6, 2, 1), plt.imshow(img, cmap='gray'), plt.title('Original Image')
plt.subplot(6, 2, 2), plt.hist(img.flatten(), bins=256, range=[0, 256]), plt.title('Original Image Histogram')

plt.subplot(6, 2, 3), plt.imshow(img_box, cmap='gray'), plt.title('Box Filter')
plt.subplot(6, 2, 4), plt.hist(img_box.flatten(), bins=256, range=[0, 256]), plt.title('Box Filter Histogram')

plt.subplot(6, 2, 5), plt.imshow(img_gaussian, cmap='gray'), plt.title('Gaussian Blur')
plt.subplot(6, 2, 6), plt.hist(img_gaussian.flatten(), bins=256, range=[0, 256]), plt.title('Gaussian Blur Histogram')

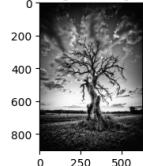
plt.subplot(6, 2, 7), plt.imshow(img_medain, cmap='gray'), plt.title('Median Blur')
plt.subplot(6, 2, 8), plt.hist(img_medain.flatten(), bins=256, range=[0, 256]), plt.title('Median Blur Histogram')

plt.subplot(6, 2, 9), plt.imshow(img_2d, cmap='gray'), plt.title('Filter 2d(function)')
plt.subplot(6, 2, 10), plt.hist(img_2d.flatten(), bins=256, range=[0, 256]), plt.title('Filter 2d(function) Histogram')

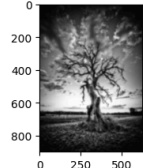
plt.subplot(6, 2, 11), plt.imshow(conv, cmap='gray'), plt.title('Filter 2d(hardcore)')
plt.subplot(6, 2, 12), plt.hist(conv.flatten(), bins=256, range=[0, 256]), plt.title('Filter 2d(hardcore) Histogram')

plt.tight_layout()
plt.show()
```

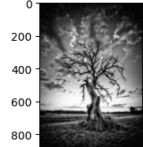
Original Image



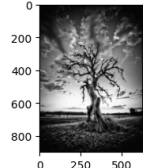
Box Filter



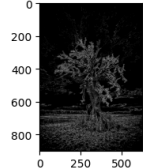
Gaussian Blur



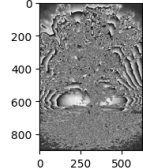
Median Blur



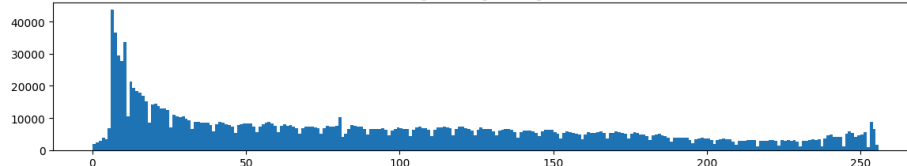
Filter 2d(function)



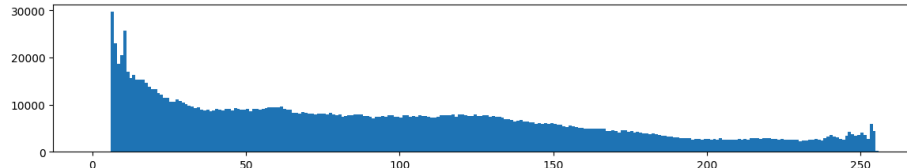
Filter 2d(hardcore)



Original Image Histogram



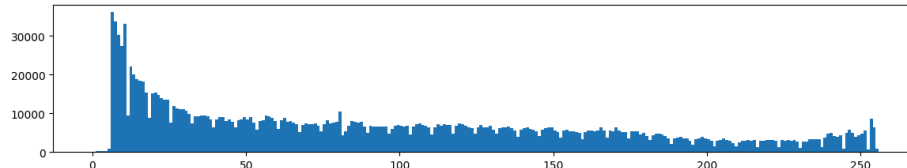
Box Filter



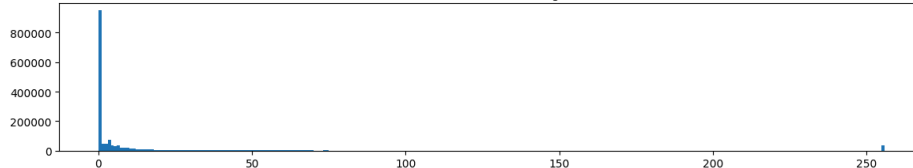
Gaussian Blur



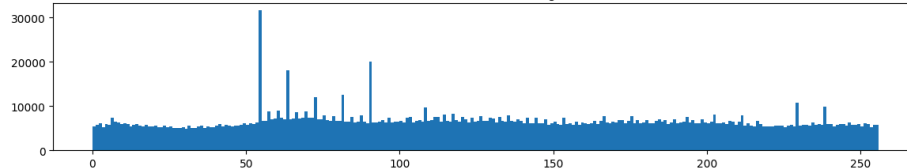
Median Blur



Filter 2d(function) Histogram



Filter 2d(hardcore) Histogram



In []:

In []: