

```

1 import cv2 as cv
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import math
5
6 def show_images(images):
7     # displaying multiple images side by side
8     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
    using-matplotlib
9
10    # err : was giving weird colormap due to diff in the mechanism of reading img of
    cv2 & matplotlib
11    # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
    matplotlib
12    # running this once in the code will ALWAYS give gray op
13    plt.gray()
14
15    n = len(images)
16    f = plt.figure()
17    for i in range(n):
18
19        # Debug, plot figure
20        axes = f.add_subplot(1, n, i + 1)
21        # the last img will show y axis on the RHS instead of LHS(which is by
    default)
22        if i==n-1:
23            axes.yaxis.tick_right()
24            plt.imshow(images[i])
25
26    plt.show(block=True)
27
28 # path to the input img
29 # path = "C:/Users/Raiyan/Desktop/img/03/Image-Processing-and-Computer-Vision-Lab/Lab
    2/Average filter/Input.png"
30 path = 'C:/Users/Raiyan/Desktop/building.jpg'
31
32 # reading img + converting from BGR to GRAY
33 img = cv.imread(path)
34 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
35
36 k_h = int(input("Enter kernel height: "))
37 k_w = k_h
38 k_size = (k_h,k_w)
39
40 # avg kernel
41 kernel = np.ones( k_size, np.float32)
42
43 # img height
44 img_h = img.shape[0]
45 # img width
46 img_w = img.shape[1]
47 #kernel height
48 a = kernel.shape[0] // 2
49 # kernel width
50 b = kernel.shape[1] // 2
51
52 # empty op img
53 output = np.zeros((img_h,img_w), np.float32)
54
55 # sum of the values of the kernel
56 k_sum = kernel.sum()
57 # print(f'ksum is {ksum}')
58
59 # visiting each pixel in the img
60 # m ta row img e ... for each row ...
61 for i in range(img_h):
62     # n ta coln img e ... for each coln ...
63     for j in range(img_w):
64
65         # empty var for calculating a summed value
66         value = 0
67         # visiting each pixel in the kernel
68         # a ta row img e ... for each row ...
69         for x in range(-a,a+1):
70             # b ta coln img e ... for each coln ...
71             for y in range(-b,b+1):
72

```

```
73         if 0 <= i+x < img_h and 0 <= j+y < img_w:  
74             value = value + kernel[a+x][b+y] * img[i+x][j+y]  
75         else:  
76             value = value + 0  
77     value = value / k_sum  
78     output[i][j] = value  
79  
80 show_images([img,output])  
81  
82
```

