

```

1 import cv2 as cv
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import math
5
6 # path to the input img
7 # path = "C:/Users/Raiyan/Desktop/img/03/Image-Processing-and-Computer-Vision-Lab/Lab
8 # 2/Average filter/Input.png"
9 path = 'C:/Users/Raiyan/Desktop/building.jpg'
10
11 # reading img + converting from BGR to GRAY
12 img = cv.imread(path)
13 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
14
15 k_h = int(input("Enter kernel height: "))
16 k_w = k_h
17 k_size = (k_h, k_w)
18
19 # avg kernel
20 kernel = np.ones( k_size, np.float32)
21
22 # img height
23 img_h = img.shape[0]
24 # img width
25 img_w = img.shape[1]
26 # kernel height // 2
27 a = kernel.shape[0] // 2
28 # kernel width // 2
29 b = kernel.shape[1] // 2
30
31 # empty op img
32 output = np.zeros((img_h, img_w), np.float32)
33
34 # sum of the values of the kernel
35 k_sum = kernel.sum()
36 # print(f'ksum is {ksum}')
37
38 # visiting each pixel in the img
39 # m ta row img e ... for each row ...
40 for i in range(img_h):
41     # n ta coln img e ... for each coln ...
42     for j in range(img_w):
43         # empty var for storing all the values
44         values = []
45         # visiting each pixel in the kernel
46         # a ta row img e ... for each row ...
47         for x in range(-a, a+1):
48             # b ta coln img e ... for each coln ...
49             for y in range(-b, b+1):
50                 if 0 <= i+x < img_h and 0 <= j+y < img_w:
51                     calculated_val = kernel[a+x][b+y] * img[i+x][j+y]
52                     values.append( calculated_val )
53                 else:
54                     values.append(0)
55             values.sort()
56
57             median_val = len(values) // 2
58
59             output[i][j] = values[median_val]
60             output[i][j] /= k_sum
61
62
63 def show_images(images):
64     # displaying multiple images side by side
65     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
66     # using-matplotlib
67     # err : was giving weird colormap due to diff in the mechanism of reading img of
68     # cv2 & matplotlib
69     # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
70     # matplotlib
71     # running this once in the code will ALWAYS give gray op
72     plt.gray()
73
74     no_of_imgs = len(images)
75     f = plt.figure()

```

```

74     for i in range(no_of_imgs):
75
76         # Debug, plot figure
77         axes = f.add_subplot(1, no_of_imgs, i + 1)
78         # the last img will show y axis on the RHS instead of LHS(which is by
default)
79
80         if i==no_of_imgs-1:
81             axes.yaxis.tick_right()
82         if i==0 :
83             plt.title('input')
84         else :
85             plt.title('op')
86             plt.imshow(images[i])
87             # plt.rc('font', size=8)
88             plt.show(block=True)
89
90 show_images([img,output])
91

```

