

```

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/building_332x317.jpg'
8
9 # reading img + converting from BGR to GRAY
10 img = cv.imread(path)
11 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
12
13 k_h = int(input("Enter kernel height: "))
14 k_w = k_h
15 k_size = (k_h,k_w)
16
17 # empty kernel
18 kernel = np.zeros( k_size, np.float32)
19
20 # img height
21 img_h = img.shape[0]
22 # img width
23 img_w = img.shape[1]
24 # kernel height // 2
25 a = kernel.shape[0] // 2
26 # kernel width // 2
27 b = kernel.shape[1] // 2
28
29 sigma = 60.0
30 normalizing_c = 1.0 / ( 2.0 * sigma * sigma * np.pi )
31
32 # building gauss kernel
33 for x in range(-a,a+1):
34     for y in range(-b,b+1):
35         dist = math.sqrt(x*x + y*y) * normalizing_c
36         val = math.exp( -dist ) * normalizing_c
37         kernel[a+x][b+y] = val
38
39 # empty op img
40 output = np.zeros((img_h,img_w), np.float32)
41
42 # conv
43 # visiting each pixel in the img
44 # m ta row img e ... for each row ...
45 for i in range(img_h):
46     # n ta coln img e ... for each coln ...
47     for j in range(img_w):
48         # sum of val to be calc
49         temp = 0
50         # visiting each pixel in the kernel
51         # a ta row img e ... for each row ...
52         for x in range(-a,a+1):
53             # b ta coln img e ... for each coln ...
54             for y in range(-b,b+1):
55                 if 0 <= i-x < img_h and 0 <= j-y < img_w:
56                     temp += kernel[a+x][b+y] * img[i-x][j-y]
57                 else:
58                     temp += 0
59         temp = temp / (k_w*k_h)
60         output[i][j] = temp
61
62
63 def show_images(images, image_title):
64     # displaying multiple images side by side
65     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-using-matplotlib
66
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-matplotlib
70     # running this once in the code will ALWAYS give gray op
71     plt.gray()
72
73     no_of_imgs = len(images)
74     f = plt.figure()
75     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
83     plt.title(image_title[i])
84     plt.imshow(images[i], 'gray')
85     # plt.rc('font', size=8)
86     plt.show(block=True)
87
88 show_images([img,output], ['input', 'output'])
89
```