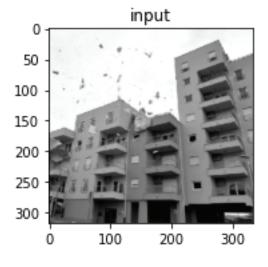
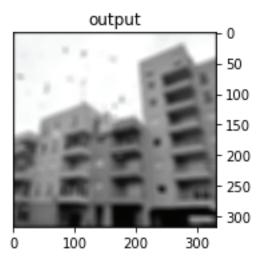
```
1 import cv2 as cv
 2 import matplotlib.pyplot as plt
 3 import numpy as np
 4 import math
 6 # path to the input img
 7 path = 'C:/Users/Raiyan/Desktop/building_332x317.jpg'
9 # reading img + converting from BGR to GRAY
10 img = cv.imread(path)
11 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
13 k h = int(input("Enter kernel height: "))
14 k_w = k_h
15 | k_size = (k_h, k_w)
17 # empty kernel
18 kernel = np.zeros( k_size, np.float32)
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
29 sigma = 60.0
30 normalizing_c = 1.0 / ( 2.0 * sigma * sigma * np.pi )
32 # building gauss kernel
33 for x in range(-a, a+1):
       for y in range(-b,b+1):
34
35
           dist = math.sqrt(x*x + y*y) * normalizing_c
           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)
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 \dots for each row \dots
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 \le i-x \le img_h and 0 \le j-y \le img_w:
                       temp += kernel[a+x][b+y] * img[i-x][j-y]
56
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):
       # displaying multiple images side by side
64
65
       # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
   using-matplotlib
66
       # err : was giving weird colormap due to diff in the mechanism of reading img of
67
   cv2 & matplotlib
68
       # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
   matplotlib
69
       # running this once in the code will ALWAYS give gray op
70
       plt.gray()
71
72
       no of imgs = len(images)
73
       f = plt.figure()
74
       for i in range(no_of_imgs):
```

5/10/22, 11:20 PM gauss.py

```
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
            if i==no_of_imgs-1:
80
                axes.yaxis.tick_right()
81
82
83
            plt.title(image_title[i])
           plt.imshow(images[i], 'gray')
# plt.rc('font', size=8)
84
85
       plt.show(block=True)
86
87
88 show_images([img,output], ['input', 'output'])
89
```





localhost:4649/?mode=python 2/2