```python
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
import math
minval = 0
maxval = 255

# path to the input img
path = 'C:/Users/Raiyan/Desktop/cube.png'

# reading img + converting from BGR to GRAY
img = cv.imread(path)
img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
# resizing image
img = cv.resize(img, (int(820/3.5),int(720/3.5)), interpolation = cv.INTER_AREA)
img = img/maxval
img1 = img

k_h = int(input("Enter kernel height: "))
k_w = k_h
k_size = (k_h,k_w)

# empty kernel
kernel = np.zeros( k_size, np.float32)

# img height
img_h = img.shape[0]
# img width
img_w = img.shape[1]
# kernel height // 2
a = kernel.shape[0] // 2
# kernel width // 2
b = kernel.shape[1] // 2

sigma = 60.0
normalizing_c = 1.0 / ( 2.0 * sigma * sigma  )

# building kernel
for x in range(-a,a+1):
    for y in range(-b,b+1):
        dist = math.sqrt(x*x + y*y) * normalizing_c
        val = math.exp( -dist ) / ( np.pi * 2.0 * sigma * sigma )
        kernel[a+x][b+y] = val

# empty op img
output = np.zeros((img_h,img_w), np.float32)

# conv
# visiting each pixel in the img
# m ta row img e ... for each row ...
for i in range(img_h):
    # n ta coln img e ... for each coln ...
    for j in range(img_w):
        # sum of val to be calc
        calc = 0
        # empty kernel for each iter
        loop_ker = np.zeros( k_size, np.float32)
        # visiting each pixel in the kernel
        # a ta row img e ... for each row ...
        for x in range(-a,a+1):
            # b ta coln img e ... for each coln ...
            for y in range(-b,b+1):
                if 0 <= i-x < img_h and 0 <= j-y < img_w:
                    dist = math.sqrt( np.power( img[i][j] - img[i-x][j-y], 2 ) ) *
normalizing_c
                    val = math.exp( -dist ) / ( np.pi * 2.0 * sigma * sigma )
                    loop_ker[a+x][b+y] = kernel[a+x][b+y] * val

        for x in range(-a,a+1):
            for y in range(-b,b+1):
                if 0 <= i-x < img_h and 0 <= j-y < img_w:
                    calc += kernel[a+x][b+y] * img[i-x][j-y]
                else:
                    calc += 0
        calc = calc / ( loop_ker.shape[0] * loop_ker.shape[1] )
        output[i][j] = calc
output *= maxval
```

```
77
78
79 def show_images(images, image_title):
80     # displaying multiple images side by side
81     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
   using-matplotlib
82
83     # err : was giving weird colormap due to diff in the mechanism of reading img of
   cv2 & matplotlib
84     # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
   matplotlib
85     # running this once in the code will ALWAYS give gray op
86     plt.gray()
87
88     no_of_imgs = len(images)
89     f = plt.figure()
90     for i in range(no_of_imgs):
91
92         # Debug, plot figure
93         axes = f.add_subplot(1, no_of_imgs, i + 1)
94         # the last img will show y axis on the RHS instead of LHS(which is by
   default)
95
96         if i==no_of_imgs-1:
97             axes.yaxis.tick_right()
98
99         plt.title(image_title[i])
100        plt.imshow(images[i])
101        # plt.rc('font', size=8)
102    plt.show(block=True)
103
104 show_images([img1,output], ['input', 'output'])
105
```