

```

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 img1 = img
13
14 # robert-x kernel
15 kernel = np.array([[1,0],
16                    [0,-1]], np.float32)
17
18 k_h = kernel.shape[0]
19 k_w = k_h
20 k_size = (k_h,k_w)
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 # conv
35 # visiting each pixel in the img
36 # m ta row img e ... for each row ...
37 for i in range(img_h):
38     # n ta coln img e ... for each coln ...
39     for j in range(img_w):
40         # sum of val to be calc
41         calc = 0
42         # visiting each pixel in the kernel
43         # a ta row img e ... for each row ...
44         for x in range(-a,a+1):
45             # b ta coln img e ... for each coln ...
46             for y in range(-b,b+1):
47                 if 0 <= i-x < img_h and 0 <= j-y < img_w:
48                     if 0<= a+x < k_h and 0<= b+y < k_w:
49                         calc += kernel[a+x][b+y] * img[i-x][j-y]
50                 else:
51                     calc += 0
52         calc = calc / (k_w*k_h)
53         output[i][j] = calc
54
55 robert_vertical = output
56 for i in range(img_h):
57     for j in range(img_w):
58         if robert_vertical[i][j] > 255:
59             robert_vertical[i][j] = 255
60         elif robert_vertical[i][j] < 0:
61             robert_vertical[i][j] = 0
62
63 output = np.zeros((img_h,img_w), np.float32)
64
65 # robert-y kernel
66 kernel = np.array([[0,1],
67                    [-1,0]], np.float32)
68
69 # conv
70 # visiting each pixel in the img
71 # m ta row img e ... for each row ...
72 for i in range(img_h):
73     # n ta coln img e ... for each coln ...
74     for j in range(img_w):
75         # sum of val to be calc
76         calc = 0
77         # visiting each pixel in the kernel

```

```

78     # a ta row img e ... for each row ...
79     for x in range(-a,a+1):
80         # b ta coln img e ... for each coln ...
81         for y in range(-b,b+1):
82             if 0 <= i-x < img_h and 0 <= j-y < img_w:
83                 if 0<= a+x < k_h and 0<= b+y < k_w:
84                     calc += kernel[a+x][b+y] * img[i-x][j-y]
85             else:
86                 calc += 0
87         calc = calc / (k_w*k_h)
88         output[i][j] = calc
89
90 robert_horizontal = output
91 for i in range(img_h):
92     for j in range(img_w):
93         if robert_horizontal[i][j] > 255:
94             robert_horizontal[i][j] = 255
95         elif robert_horizontal[i][j] < 0:
96             robert_horizontal[i][j] = 0
97
98 robert_merged = robert_horizontal + robert_vertical
99 for i in range(img_h):
100     for j in range(img_w):
101         if robert_merged[i][j] > 255:
102             robert_merged[i][j] = 255
103         elif robert_merged[i][j] < 0:
104             robert_merged[i][j] = 0
105
106 img = img + robert_merged
107
108 for i in range(img_h):
109     for j in range(img_w):
110         if img[i][j] > 255:
111             img[i][j] = 255
112         elif img[i][j] < 0:
113             img[i][j] = 0
114
115 def show_images(images, image_title):
116     # displaying multiple images side by side
117     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
    using-matplotlib
118
119     # err : was giving weird colormap due to diff in the mechanism of reading img of
    cv2 & matplotlib
120     # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
    matplotlib
121     # running this once in the code will ALWAYS give gray op
122     plt.gray()
123
124     no_of_imgs = len(images)
125     f = plt.figure()
126     for i in range(no_of_imgs):
127
128         # Debug, plot figure
129         axes = f.add_subplot(1, no_of_imgs, i + 1)
130         # the last img will show y axis on the RHS instead of LHS(which is by
    default)
131
132         if i==no_of_imgs-1:
133             axes.yaxis.tick_right()
134
135         plt.title(image_title[i])
136         plt.imshow(images[i], 'gray')
137         # plt.rc('font', size=8)
138         plt.show(block=True)
139
140 show_images([img1,robert_vertical],
141             ['input', 'robert vertical'])
142 show_images([robert_horizontal, robert_merged],
143             ['robert horizontal', 'robert merged'])
144 show_images([img1,img],
145             ['input', 'enhanced output'])
146
147

```

