

```

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 # scharr-y kernel
15 kernel = np.array([[-3,0,3],
16                   [-10,0,10],
17                   [-3,0,3]], np.float32)
18
19 k_h = kernel.shape[0]
20 k_w = k_h
21 k_size = (k_h,k_w)
22
23 # img height
24 img_h = img.shape[0]
25 # img width
26 img_w = img.shape[1]
27 # kernel height // 2
28 a = kernel.shape[0] // 2
29 # kernel width // 2
30 b = kernel.shape[1] // 2
31
32 # empty op img
33 output = np.zeros((img_h,img_w), np.float32)
34
35 # conv
36 # visiting each pixel in the img
37 # m ta row img e ... for each row ...
38 for i in range(img_h):
39     # n ta coln img e ... for each coln ...
40     for j in range(img_w):
41         # sum of val to be calc
42         calc = 0
43         # visiting each pixel in the kernel
44         # a ta row img e ... for each row ...
45         for x in range(-a,a+1):
46             # b ta coln img e ... for each coln ...
47             for y in range(-b,b+1):
48                 if 0 <= i-x < img_h and 0 <= j-y < img_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 scharr_vertical = output
56 for i in range(img_h):
57     for j in range(img_w):
58         if scharr_vertical[i][j] > 255:
59             scharr_vertical[i][j] = 255
60         elif scharr_vertical[i][j] < 0:
61             scharr_vertical[i][j] = 0
62
63 output = np.zeros((img_h,img_w), np.float32)
64
65 # scharr-x kernel
66 kernel = np.array([[-3,-10,-3],
67                   [0,0,0],
68                   [3,10,3]], np.float32)
69
70 # conv
71 # visiting each pixel in the img
72 # m ta row img e ... for each row ...
73 for i in range(img_h):
74     # n ta coln img e ... for each coln ...
75     for j in range(img_w):
76         # sum of val to be calc
77         calc = 0

```

```

78     # visiting each pixel in the kernel
79     # a ta row img e ... for each row ...
80     for x in range(-a,a+1):
81         # b ta coln img e ... for each coln ...
82         for y in range(-b,b+1):
83             if 0 <= i-x < img_h and 0 <= j-y < img_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 scharr_horizontal = output
91 for i in range(img_h):
92     for j in range(img_w):
93         if scharr_horizontal[i][j] > 255:
94             scharr_horizontal[i][j] = 255
95         elif scharr_horizontal[i][j] < 0:
96             scharr_horizontal[i][j] = 0
97
98 scharr_merged = scharr_horizontal + scharr_vertical
99 for i in range(img_h):
100     for j in range(img_w):
101         if scharr_merged[i][j] > 255:
102             scharr_merged[i][j] = 255
103         elif scharr_merged[i][j] < 0:
104             scharr_merged[i][j] = 0
105
106 img = img + scharr_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
116 def show_images(images, image_title):
117     # displaying multiple images side by side
118     # https://stackoverflow.com/questions/41793931/plotting-images-side-by-side-
    using-matplotlib
119
120     # err : was giving weird colormap due to diff in the mechanism of reading img of
    cv2 & matplotlib
121     # https://stackoverflow.com/questions/3823752/display-image-as-grayscale-using-
    matplotlib
122     # running this once in the code will ALWAYS give gray op
123     plt.gray()
124
125     no_of_imgs = len(images)
126     f = plt.figure()
127     for i in range(no_of_imgs):
128
129         # Debug, plot figure
130         axes = f.add_subplot(1, no_of_imgs, i + 1)
131         # the last img will show y axis on the RHS instead of LHS(which is by
    default)
132
133         if i==no_of_imgs-1:
134             axes.yaxis.tick_right()
135
136         plt.title(image_title[i])
137         plt.imshow(images[i], 'gray')
138         # plt.rc('font', size=8)
139         plt.show(block=True)
140
141 show_images([img1,scharr_vertical],
142             ['input', 'scharr vertical'])
143 show_images([scharr_horizontal, scharr_merged],
144             ['scharr horizontal', 'scharr merged'])
145 show_images([img1,img],
146             ['input', 'enhanced output'])
147
148

```

