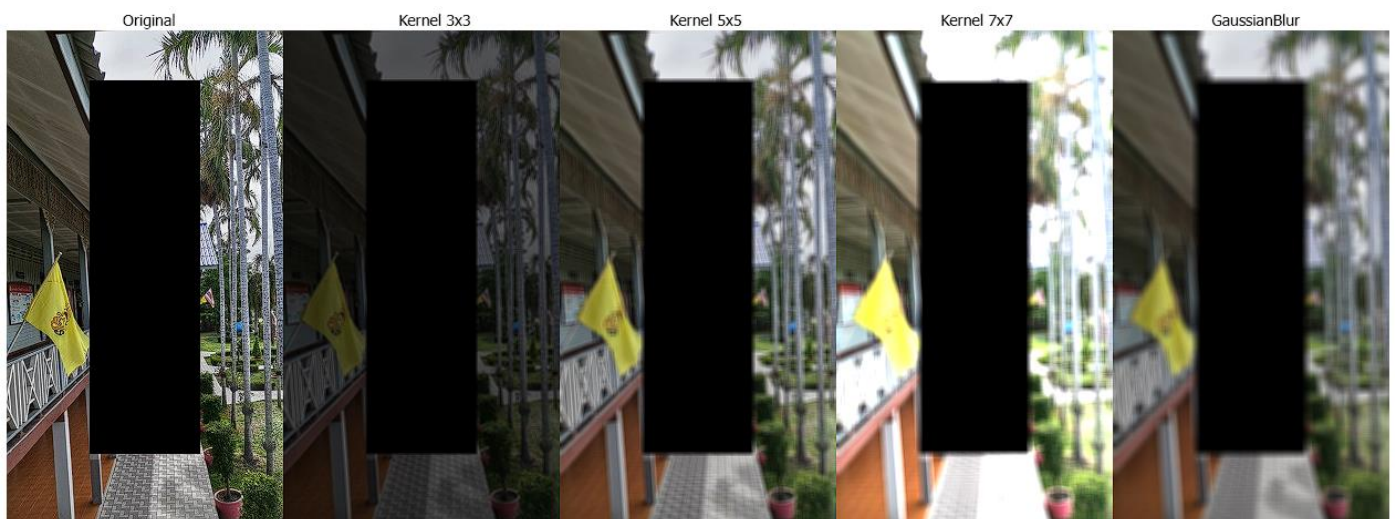


Region Blurred

Code:

```
1 # Try your own applying Averaging Filter
2 # instead of using the given cv.blur()
3
4 # Averaging/smooth/blur mask of different sizes, i.e., 3x3, 5x5, 7x7 and Gaussian blur
5
6 kernel5 = np.array([
7     [ 1, 1, 1, 1, 1],
8     [ 1, 1, 1, 1, 1],
9     [ 1, 1, 1, 1, 1],
10    [ 1, 1, 1, 1, 1],
11    [ 1, 1, 1, 1, 1],
12    ])
13 kernel3 = np.array([
14    [ 1, 1, 1],
15    [ 1, 1, 1],
16    [ 1, 1, 1],
17    ])
18 kernel7 = np.array([
19    [ 1, 1, 1, 1, 1, 1, 1],
20    [ 1, 1, 1, 1, 1, 1, 1],
21    [ 1, 1, 1, 1, 1, 1, 1],
22    [ 1, 1, 1, 1, 1, 1, 1],
23    [ 1, 1, 1, 1, 1, 1, 1],
24    [ 1, 1, 1, 1, 1, 1, 1],
25    [ 1, 1, 1, 1, 1, 1, 1],
26    ])
27 kernel5.shape
28 print("\t\t\t\tOriginal \t\t\t\t\t Kernel 3x3 \t\t\t\t\t Kernel 5x5 \t\t\t\t\t Kernel 7x7 \t\t\t\t\t GaussianBlur")
29 filtered5_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel5)
30 filtered3_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel3)
31 filtered7_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel7)
32 # when depth=-1, the output image will have the same depth as the source.
33
34 res = np.hstack((cut_out, filtered3_image, filtered5_image, filtered7_image, img_blur2))
35 cv2_imshow(res)
36 #print('Smoothed/blurred image ', filtered5_image.shape)
37
38 # Now try a different smoothing mask, e.g., a Gaussian blur
39
40 # After doing all that, do you find any differences? Provide your explanations.
41
```

Output:



จากการทำ Blur Masking ด้วย kernel ขนาดต่างกันตั้งแต่ 3×3 , 5×5 , 7×7 เปรียบเทียบกับภาพต้นฉบับ และภาพที่ถูกเบลอด้วย Gaussian Blur พบว่ายิ่ง mask มีขนาดใหญ่ขึ้น ภาพยิ่งมีค่าสีที่สูงขึ้น (ภาพสว่างขึ้น) เนื่องจาก mask เป็นการนำค่า pixel โดยรอบของ pixel นั้นๆ มารวมกันแล้ว filter ด้วยการนำ pixel หารด้วยค่าคงที่ (code line 29 - 31) คือ หารด้วย 25 เป็นการ weight ค่า pixel ใหม่ที่เท่ากัน เมื่อสังเกตดูจะพบว่า masking 3×3 จะทำให้ภาพค่าสีต่ำลง (ภาพมืดลง) masking 7×7 จะทำให้ภาพค่าสีสูงขึ้น (ภาพสว่างขึ้น) แต่ masking 5×5 จะได้ภาพที่ค่าสีใกล้เคียงภาพต้นฉบับ (ภาพสว่างเท่าภาพต้นฉบับ) เพราะว่า เมื่อหารด้วย 25 ค่าจากจำนวน pixel ที่เหมาะสมจึงเป็น 25 pixel เช่นกัน masking 5×5 จึงได้ผลลัพธ์ที่สีภาพใกล้เคียงต้นฉบับมากที่สุด

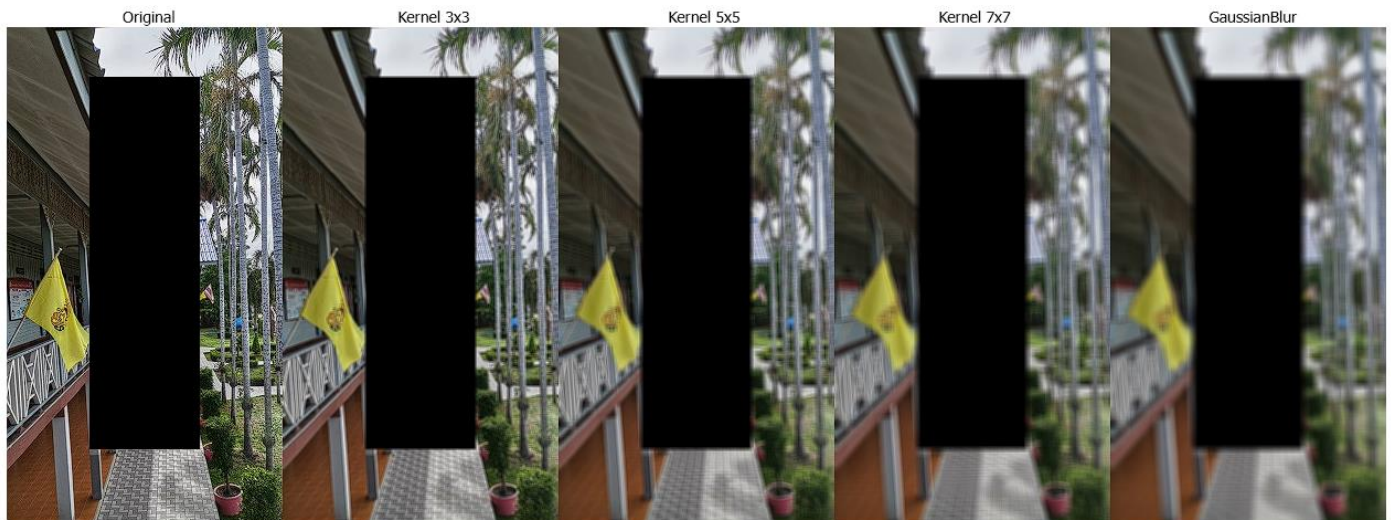
ในทำนองเดียวกัน หาก filter แทนที่เลข 25 ด้วยเลขค่าอื่น เช่น 9 masking ที่เหมาะสมจะเป็นขนาด 3×3 ได้ผลลัพธ์ดังรูปต่อไปนี้

```
29 filtered5_image = cv.filter2D(cut_out, -1, (1/9)*kernel5)
30 filtered3_image = cv.filter2D(cut_out, -1, (1/9)*kernel3)
31 filtered7_image = cv.filter2D(cut_out, -1, (1/9)*kernel7)
```



เมื่อปรับค่า filter ให้เหมาะสมตามขนาดของ masking (masking 3*3 กับ filter 1/9, masking 5*5 กับ filter 1/25, masking 7*7 กับ filter 1/49) จะพบว่า ยิ่ง masking มีขนาดใหญ่ขึ้น ภาพจะยิ่งเบลอมากขึ้น เนื่องจากการนำค่า pixel โดยรอบมาเฉลี่ยกัน ทำให้แต่ละ pixel นั้นๆ มีความใกล้เคียงกัน สีสันกันมากยิ่งขึ้น

```
29 filtered5_image = cv.filter2D(cut_out, -1, (1/25)*kernel5)
30 filtered3_image = cv.filter2D(cut_out, -1, (1/9)*kernel3)
31 filtered7_image = cv.filter2D(cut_out, -1, (1/49)*kernel7)
```



อ้างอิงจาก Document ของ Open CV ในส่วนของ Gaussian Blur จะเป็น filter ที่ weight ค่า pixel ใหม่ไม่เท่ากัน โดยให้ pixel ตรงกลางมี weight ที่สูงที่สุด ภาพผลลัพธ์จึงมีความสว่างใกล้เคียงกับภาพต้นฉบับ