

## Region Blurred

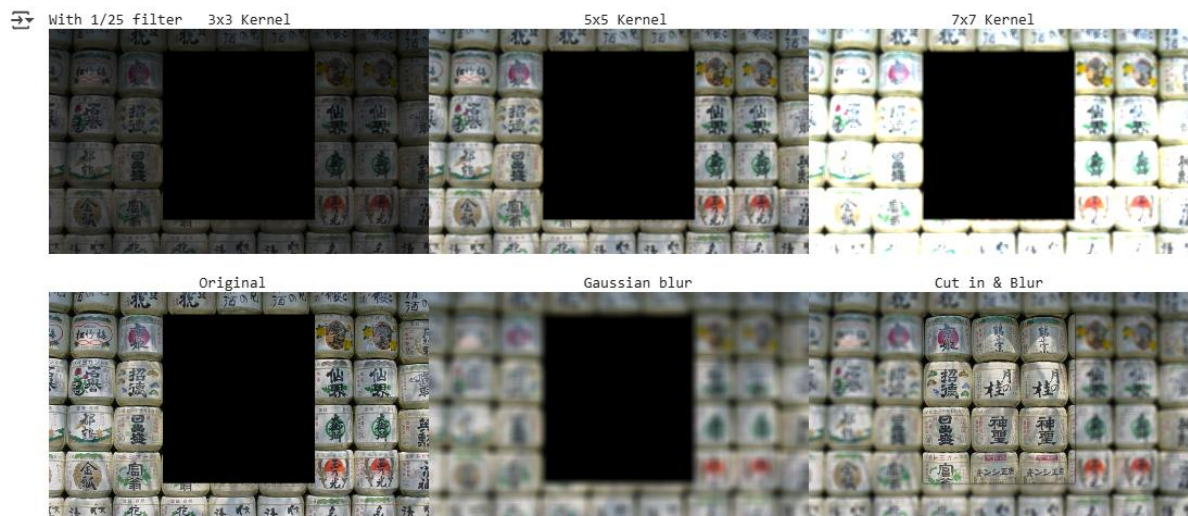
Code ที่ใช้

```
1 kernel_3 = np.array([
2     [1, 1, 1],
3     [1, 1, 1],
4     [1, 1, 1]
5 ])
6
7 kernel_5 = np.array([
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    [1, 1, 1, 1, 1]
13 ])
14
15 kernel_7 = np.array([
16    [1, 1, 1, 1, 1, 1, 1],
17    [1, 1, 1, 1, 1, 1, 1],
18    [1, 1, 1, 1, 1, 1, 1],
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 ])
24
25 print(f"Kernel size = 3x3: {kernel_3.shape}, 5x5: {kernel_5.shape}, 7x7: {kernel_7.shape}")
```

Kernel size = 3x3: (3, 3), 5x5: (5, 5), 7x7: (7, 7)

```
[63] 1 filtered_3x3_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel_3)
2     filtered_5x5_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel_5)
3     filtered_7x7_image = cv.filter2D(cut_out, -1, (1/25.0)*kernel_7)
4     # when depth=-1, the output image will have the same depth as the source.
5
6     resize_value = (350, 207)
7     rs_cutout = cv.resize(cut_out, resize_value)
8     rs_3x3 = cv.resize(filtered_3x3_image, resize_value)
9     rs_5x5 = cv.resize(filtered_5x5_image, resize_value)
10    rs_7x7 = cv.resize(filtered_7x7_image, resize_value)
11    img_gblur = cv.GaussianBlur(rs_cutout, (9,9), 5)
12    rs_cutin = cv.resize(img_final, resize_value)
13
14    res1 = np.hstack((rs_3x3, rs_5x5, rs_7x7))
15    res2 = np.hstack((rs_cutout, img_gblur, rs_cutin))
16
17    print('With 1/25 filter   3x3 Kernel\t\t\t5x5 Kernel\t\t\t7x7 Kernel')
18    cv2_imshow(res1)
19    print('')
20    print('\t\t Original\t\t\tGaussian blur\t\t\t Cut in & Blur')
21    cv2_imshow(res2)
```

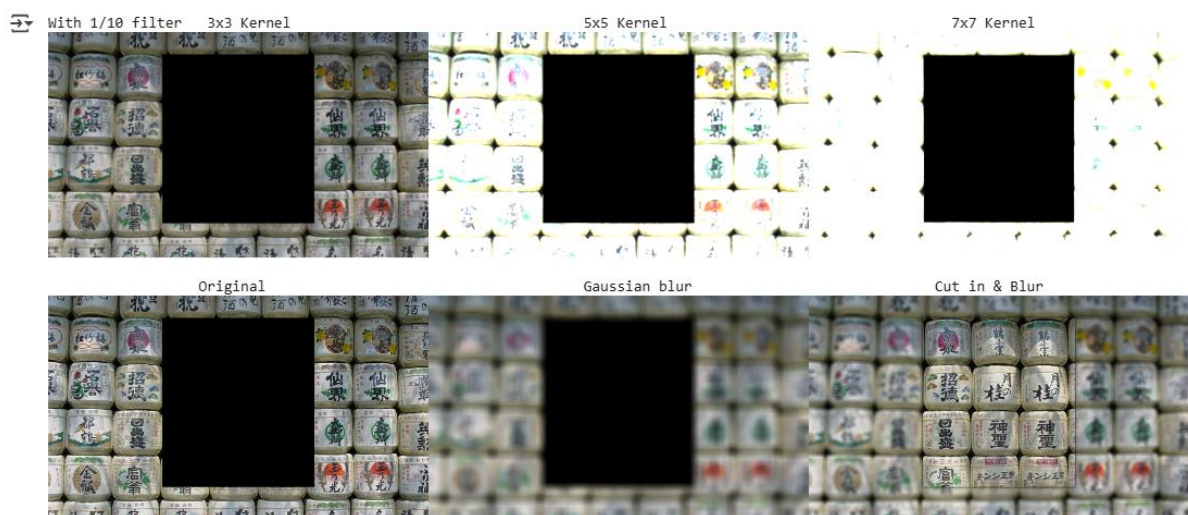
ผลที่ได้



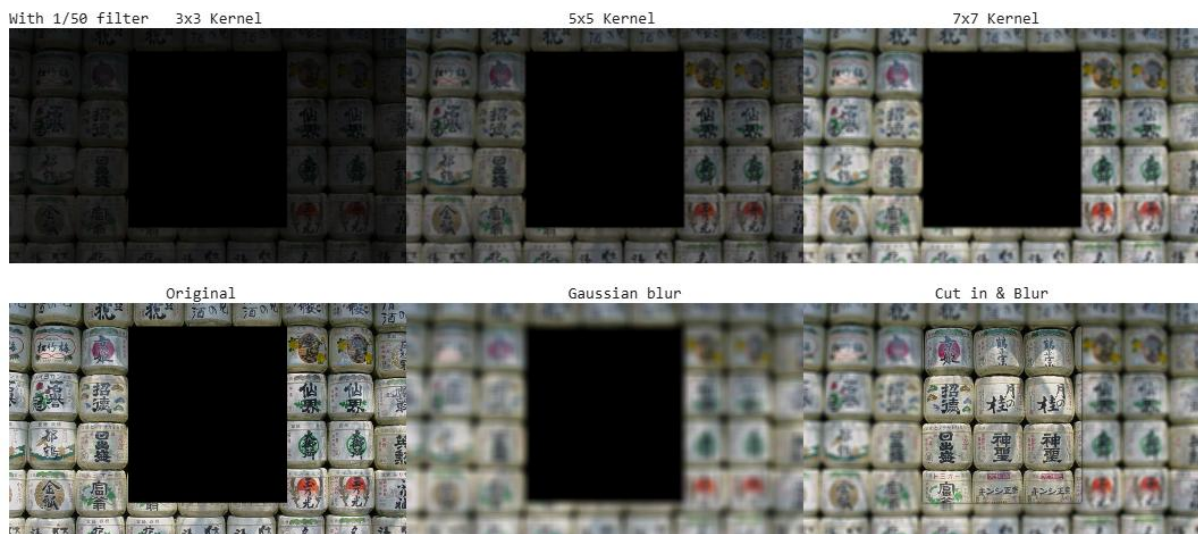
จากผลที่ออกมาจะพบว่าการใช้สูตร  $1/25$  จะได้ผลที่ไม่เหมือนกันสำหรับ kernel ในแต่ละขนาด ซึ่ง kernel ยิ่งมีขนาดที่ใหญ่ขึ้นภาพจะยิ่งมีค่าที่ใกล้เคียงขาวมากขึ้น สาเหตุเพราะขนาดของ kernel เป็นตัวกำหนดจุดของ pixel ที่จะมาทำการรวมกันแล้วแล้วมาหารด้วยค่าที่เรากำหนดไว้ ซึ่งจะแตกต่างกับภาพที่ถูกเบลอด้วยเทคนิค Gaussian blur ที่ในสูตรจะมีการคำนวณและ weight ค่าให้เหมาะสมในแต่ละพื้นที่

หากลองปรับค่าคงที่ในการทำ filter ให้เป็น  $1/10$  และ  $1/50$  ผลที่ได้จะเป็นดังรูปถัดไป

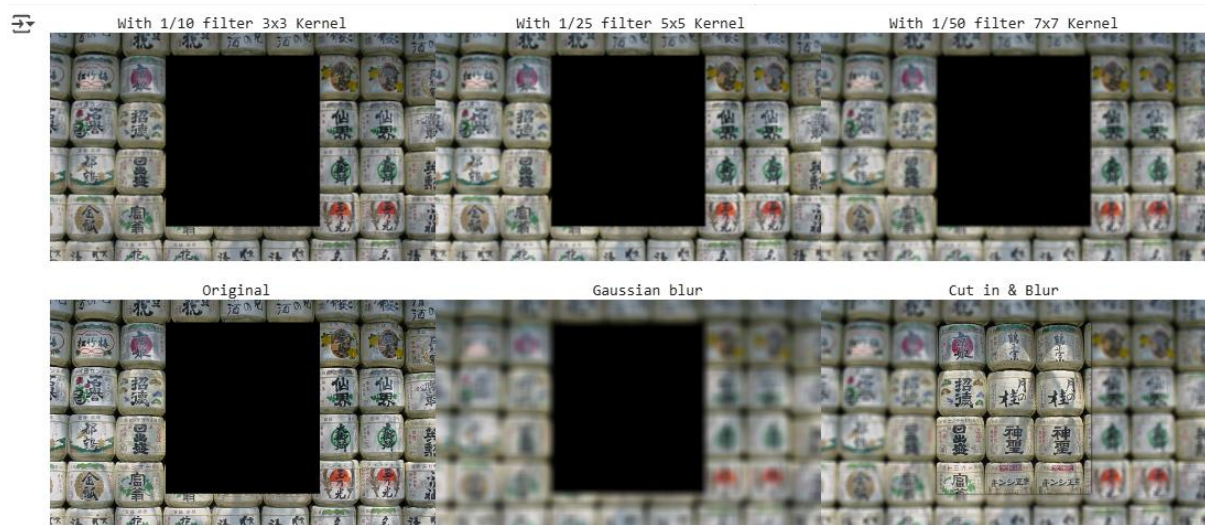
ค่าคงที่ =  $1/10$



ค่าคงที่ =  $1/50$



ผลที่ได้จากการเปลี่ยนค่าคงที่ทำให้ทราบว่า ยิ่งค่าที่หารกับ 1 ยิ่งน้อยเท่าไรภาพจะยิ่งสว่าง(ค่าที่หารออกมาเข้าใกล้ 1) และค่าที่หารกับ 1 ยิ่งมากเท่าไรภาพจะมีดขึ้น(ค่าที่หารออกมาเข้าใกล้ 0) เราจึงควรตั้งขนาด kernel และค่าคงตัวให้สัมพันธ์กันเพื่อหลีกเลี่ยงภาพที่สว่างหรือมืดเกินไป



หลังจากเลือกค่าคงตัวที่เหมาะสมกับแต่ละ kernel แล้วจะพบว่า kernel ยิ่งขนาดใหญ่เท่าไรภาพจะมีความเบลอมามากขึ้นเท่านั้นเนื่องจาก kernel ยิ่งกว้างการเฉลี่ยค่าจาก pixel รอบด้านยิ่งเยอะเท่านั้น