


Lab 7 Salt&Pepper / Gaussian Noise

1. นำภาพมาเติม Noise Salt & Pepper & Gaussian noise

```
import cv2 as cv
import numpy as np
import random
from matplotlib import pyplot as plt

original_img = cv.imread('output.png')
RGB_img = cv.cvtColor(original_img, cv.COLOR_BGR2RGB)
plt.rcParams["figure.figsize"] = [5, 8]
plt.imshow(RGB_img)
plt.show()
```



```
# modified from https://gist.github.com/Prasad9/28f6a2df8e8d463c6ddd040f4f6a028a
def gauss_noise(img, mean = 0, var = 200):
    sigma = var ** 0.5
    gaussian1 = np.random.normal(mean, sigma, (img.shape[0],img.shape[1]))
    gaussian2 = np.random.normal(mean, sigma, (img.shape[0],img.shape[1]))
    gaussian3 = np.random.normal(mean, sigma, (img.shape[0],img.shape[1]))
    noisy_image = np.zeros(img.shape, np.float32)
    if len(img.shape) == 2:
        noisy_image = img + gaussian
    else:
        noisy_image[:, :, 0] = img[:, :, 0] + gaussian1
        noisy_image[:, :, 1] = img[:, :, 1] + gaussian2
        noisy_image[:, :, 2] = img[:, :, 2] + gaussian3
    cv.normalize(noisy_image, noisy_image, 0, 255, cv.NORM_MINMAX, dtype=-1)
    noisy_image = noisy_image.astype(np.uint8)
    return noisy_image
```

รหัสนักศึกษา 162404140008-2

ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

```
# modified from https://www.py4u.net/discuss/14307
def sp_noise(img, prob=0.05):
    output = np.zeros(img.shape, np.uint8)
    thres = 1 - prob
    ch = 3 if len(output.shape) != 2 else 1
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            rdn = random.random()
            if rdn < prob:
                output[i][j] = (0,0,0) if ch==3 else 0 #pepper
            elif rdn > thres:
                output[i][j] = (255,255,255) if ch==3 else 255 #salt
            else:
                output[i][j] = img[i][j]
    return output
```

✓ 0.4s Python


Generate Salt and Pepper Noise

```
sp_img = sp_noise(RGB_img, 0.02)
gauss_img = gauss_noise(RGB_img)
```

✓ 0.7s Python

```
plt.rcParams["figure.figsize"] = [15, 7]
plt.subplot(131), plt.imshow(RGB_img, interpolation='none'), plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(132), plt.imshow(sp_img, interpolation='none'), plt.title('AddedSalt&Pepper'), plt.xticks([]), plt.yticks([])
plt.subplot(133), plt.imshow(gauss_img, interpolation='none'), plt.title('AddedGaussianNoise'), plt.xticks([]), plt.yticks([])
plt.show()
```

✓ 0.1s Python



2. ทำการลด noise ที่เกิดขึ้น

2.1. ลด Noise Salt & Pepper

2.1.1. Code

```
# แก่ไลยนะ อีอิ แก่ รูปในตัวแปร sp_img ให้เหมือนเดิมมากที่สุด
median = cv.medianBlur(sp_img, 3)
compare = np.concatenate((sp_img, median), axis=1) #side by side comparison
```

✓ 0.4s

```
# รันผลลัพธ์ให้ดูด้วยนะ อีอิ เปรียบเทียบก่อนแก้ ในตัวแปร sp_img และหลังแก้ในตัวแปรที่เราเก็บภาพไว้เปรียบเทียบกับ
```

```
plt.imshow(compare)
plt.show()
```

✓ 0.3s

2.1.2. Result



2.1.3. คำอธิบายผลลัพธ์ แสกนภาพทั้งหมดโดยใช้ Median filter 3x3 และคำนวณหาค่าตรงกลางใหม่ โดยการหาค่ามัธยฐานของค่าทั้งหมด ยกตัวอย่าง [22, 22, 23, 24, 27, 27, 29, 31, 108] ค่ามัธยฐานของเซตนี้คือ 27.

2.2. ลด Gaussian Noise

2.2.1. Code

จงแก้ Noise Gaussian

```
# แก้ไลเยนนะ อีอี แก้ รูปในตัวแปร gauss_img ให้เหมือนเดิมมากที่สุด
dst = cv.fastNlMeansDenoisingColored(gauss_img, None, 10, 10, 7, 21)
```

✓ 0.2s

```
# รันผลลัพธ์ให้ดูด้วยนะ อีอี เปรียบเทียบก่อนแก้ ในตัวแปร gauss_img และหลังแก้ในตัวแปรที่เราเก็บภาพไว้เปรียบเทียบกับ
plt.subplot(121), plt.imshow(gauss_img)
plt.subplot(122), plt.imshow(dst)
plt.show()
```

✓ 0.5s

2.2.2. Result



2.2.3. คำอธิบายผลลัพธ์ หาชุดของรูปภาพที่คล้าย ๆ กัน แล้วหาค่าเฉลี่ยของหน้าตาทั้งหมด จากนั้นแทนที่ pixel ในภาพด้วยผลลัพธ์ค่าเฉลี่ยที่หาได้