Denoising image

```
1   import cv2 as cv
2   import numpy as np
3   import random
```

1.ใช้ Library random เพื่อrandomค่าสร้าง salt&pepper noise

```
#function วัดความเหมือนกันของภาพ
def compute_ssim(img1, img2):
    mean1 = np.mean(img1)
    mean2 = np.mean(img2)
    std1 = np.std(img1)
    std2 = np.std(img2)
    cov = np.cov(img1.flatten(), img2.flatten())[0, 1]

    c1 = (0.01 * 255) ** 2
    c2 = (0.03 * 255) ** 2

    ssim = (
        (2 * mean1 * mean2 + c1)
        * (2 * cov + c2)
        / ((mean1 ** 2 + mean2 ** 2 + c1) * (std1 ** 2 + std2 ** 2 + c2))
    )

    return ssim
```

2.สร้าง function ด้วยสมการ SSIM

```
img = cv.imread('golden.jpg', cv.IMREAD_GRAYSCALE)
imgOrigin = cv.imread('golden.jpg', cv.IMREAD_GRAYSCALE)
```

3.อ่านรูป และอีกอันเก็บเป็นรูปดั้งเดิม

```
#ขนาดnoise
density_salt = 0.1
density_pepper = 0.1
```

4.สร้างขนาดของ salt&pepper noise

```
# Set salt
number_of_white_pixel = int(density_salt * (img.shape[0] * img.shape[1]))
```

```
# Set pepper
number_of_black_pixel = int(density_pepper * (img.shape[0] * img.shape[1]))
```

5.ตั้งค่าให้ salt&pepper มีขนาดเป็น 10% ของรูป

```
# Add salt
for i in range(number_of_white_pixel):
    x_coord = random.randint(0, img.shape[1] - 1)
    y_coord = random.randint(0, img.shape[0] - 1)
    img[y_coord, x_coord] = 255
```

```
# Add pepper
for i in range(number_of_black_pixel):
    x_coord = random.randint(0, img.shape[1] - 1)
    y_coord = random.randint(0, img.shape[0] - 1)
    img[y_coord, x_coord] = 0
```

6.ใส่ salt&pepper ลงไปในรูปตั้งแต่ตำแหน่ง pixel แรกถึงสุดท้ายทั้งแนวตั้งและนอน

```
output = img
maxSSIM = 0
blur = 1  #ต้องเป็นเลขแรก(คี่)
```

7.เก็บค่า img ไว้ใน output เพื่อไม่ให้ซ้ำกัน และ ตั้งให้ SSIM สูงสุดยังไม่มีค่า และกำหนดให้ blur เป็นค่าแรกโดยใช้ได้เฉพาะเลขคี่

```
for i in range (1,19,2):
  #Add midblur for check (test)
  img = cv.medianBlur(img,i)
  compare = compute_ssim(imgOrigin,img)
  print(compare*100)

  #ดูค่า SSIM มากที่สุด
  if compare > maxSSIM:
    maxSSIM = compare
    blur = i

    print("Max SSIM : ", maxSSIM*100," Blur :", blur )
```

8.ใช้ for loop ในการทดสอบค่าเปรียบเทียบของการ blur ด้วยสมการ SSIM แล้วนำมาคูณ 100 เพื่อคิดเป็น % จากนั้น ใช้ if ที่อยู่ใน for loop เพื่อเก็บเฉพาะค่าที่มีเปอร์เซ็นสูงที่สุดและค่า blur ของ เปอร์เซ็นนั้น

```
#Real midblur
output = cv.medianBlur(output,blur)
```

9. แล้วจึงนำ ค่า blur ที่สูงที่สุดที่ได้มาใช้จริงโดยเก็บไว้ที่ output

```
cv.imshow('image_without_sp', output)
cv.imwrite('image_without_sp.png', output)
cv.waitKey(0)
cv.destroyAllWindows()
```

10.แสดงผลที่ได้และเก็บไว้ในไฟล์ .png