รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

Lab 8 Sobel / Lapacian / Canny Edge Detection

1. ให้นศ.นำภาพที่มีขอบทั้งในแนวตั้งและแนวนอนชัดเจน Load ภาพเข้าไปใส่ใน numpy array

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
```
✓ 0.5s

```
original_img = cv.imread('output.png')
RGB_img = cv.cvtColor(original_img, cv.COLOR_BGR2RGB)
plt.imshow(RGB_img)
plt.show()
```
✓ 0.3s

---

2. สร้างฟังก์ชันหาของด้วยวิธีการของ Sobel โดยหาขอบทั้งแกน x y และนำมารวมกันโดยฟังก์ชัน addWeight ด้วยสัดส่วนขอบในแกนxและy อย่างละ 0.5 หลังจากนั้นให้เรียกใช้งานฟังก์ชันเพื่อหาของและแสดงผลภาพต้นฉบับ ภาพผลลัพธ์ขอบในแกนx y และภาพขอบรวมทั้งในแกนx y

Sobel Edge Detector

```
def sobelDetector(img, scale = 1, delta = 0, ddepth = cv.CV_16S):
    src = img.copy()
    src = cv.GaussianBlur(src, (3, 3), 0)
    gray = cv.cvtColor(src, cv.COLOR_RGB2GRAY)
    grad_x = cv.Sobel(gray, ddepth, 1, 0, ksize=3, scale=scale, delta=delta, borderType=cv.BORDER_DEFAULT)
    # Gradient-Y
    # grad_y = cv.Scharr(gray,ddepth,0,1)
    grad_y = cv.Sobel(gray, ddepth, 0, 1, ksize=3, scale=scale, delta=delta, borderType=cv.BORDER_DEFAULT)
    abs_grad_x = cv.convertScaleAbs(grad_x)
    abs_grad_y = cv.convertScaleAbs(grad_y)
    grad = cv.addWeighted(abs_grad_x, 0.5, abs_grad_y, 0.5, 0)
    return grad,abs_grad_x,abs_grad_y
```
✓ 0.4s Python

```
sobel_img,sobel_x,sobel_y = sobelDetector(RGB_img)
```
✓ 0.4s Python

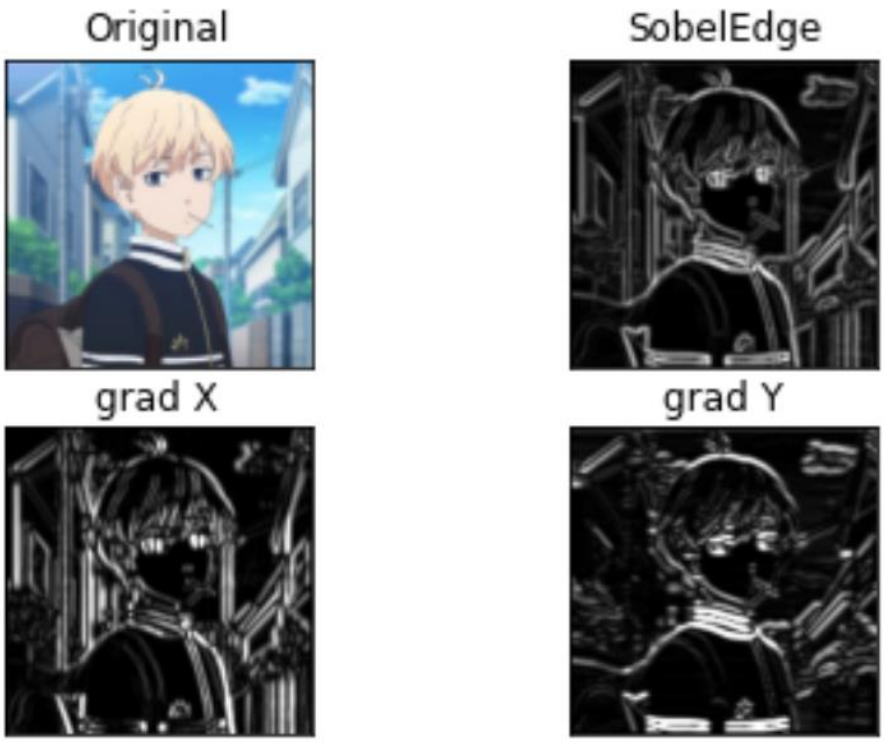
รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

```
plt.subplot(221),plt.imshow(RGB_img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(222),plt.imshow(sobel_img,cmap='gray'),plt.title('SobelEdge')
plt.xticks([]), plt.yticks([])
plt.subplot(223),plt.imshow(sobel_x,cmap='gray'),plt.title('grad X')
plt.xticks([]), plt.yticks([])
plt.subplot(224),plt.imshow(sobel_y,cmap='gray'),plt.title('grad Y')
plt.xticks([]), plt.yticks([])
plt.show()
```

✓ 0.2s

- ผลลัพธ์



รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

3. **สร้างฟังก์ชันหาของด้วยวิธีการของ Lapacian หลังจากนั้นให้เรียกใช้งานฟังก์ชันเพื่อหาของและแสดงผลภาพต้นฉบับ ภาพผลลัพธ์จากการหาขอบของ Lapacian**

Lapacian Edge Detector

```
def laplaceDetector(img,kernel_size = 3,ddepth = cv.CV_16S):
    src = img.copy()
    src = cv.GaussianBlur(src, (3, 3), 0)
    src_gray = cv.cvtColor(src, cv.COLOR_RGB2GRAY)
    dst = cv.Laplacian(src_gray, ddepth, ksize=kernel_size)
    abs_dst = cv.convertScaleAbs(dst)
    return abs_dst
```

✓ 0.2s

```
laplace_image = laplaceDetector(RGB_img,3)
```

✓ 0.3s

```
plt.subplot(121),plt.imshow(RGB_img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(laplace_image,cmap='gray'),plt.title('LaplaceEdge')
plt.xticks([]), plt.yticks([])
plt.show()
```

✓ 0.1s

- ผลลัพธ์

รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

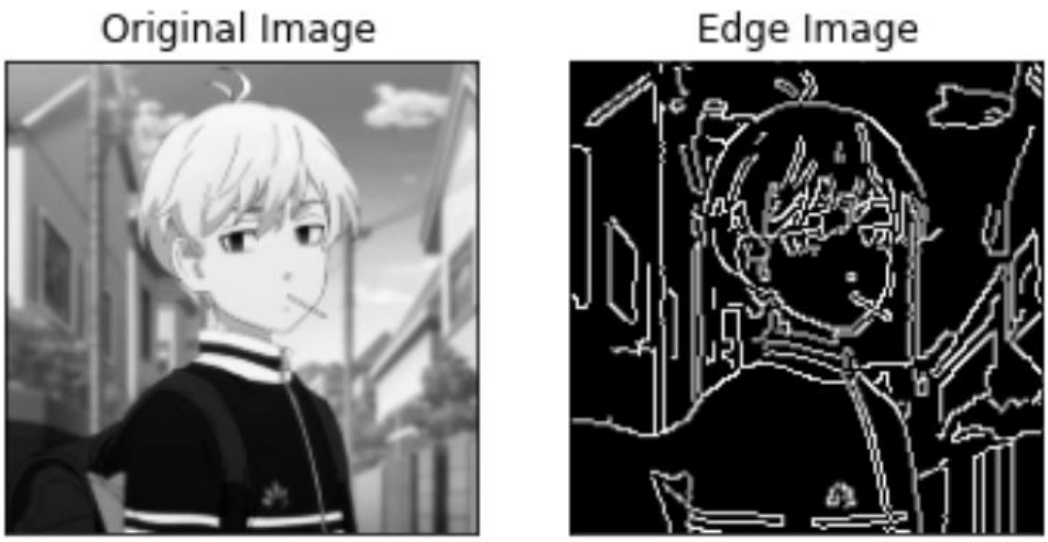
4. เรียกฟังก์ชันหาขอบด้วยวิธีการของ Canny โดยปรับค่า low_threshold / high_threshold ให้สามารถดึงขอบมาได้มากที่สุด และแสดงผลภาพต้นฉบับ ภาพผลลัพธ์จากการหาขอบของ Canny

Canny Detector

```
img = cv.imread('output.png',0)
edges = cv.Canny(img,100,200)
plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([]), plt.yticks([])
plt.show()
```

✓ 0.1s

- ผลลัพธ์



รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 27/09/2564

5. แสดงผลภาพต้นฉบับ เปรียบเทียบกับ Sobel vs Lapacian vs Canny

show all

```
plt.subplot(221),plt.imshow(RGB_img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(222),plt.imshow(laplace_image,cmap='gray'),plt.title('LaplaceEdge')
plt.xticks([]), plt.yticks([])
plt.subplot(223),plt.imshow(sobel_img,cmap='gray'),plt.title('SobelEdge')
plt.xticks([]), plt.yticks([])
plt.subplot(224),plt.imshow(edges,cmap = 'gray')
plt.title('Canny'), plt.xticks([]), plt.yticks([])
plt.show()
```

✓ 0.2s

- ผลลัพธ์