Computer vision – HW7

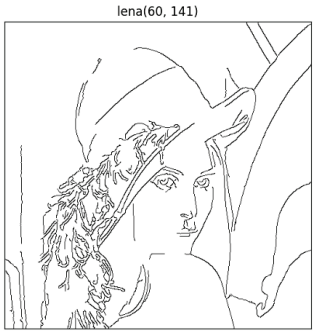
A、Source code

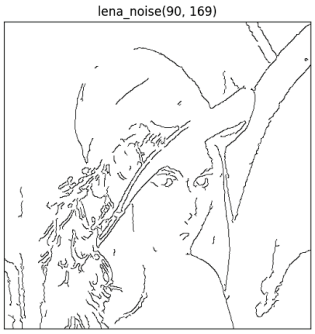
```
# WeiWen Wu 2023-11-01 15:23:19
import cv2 as cv
from matplotlib import pyplot as plt
from numpy import ndarray   # type

def bgr2gray(img_file:str) -> ndarray:
    """Convert bgr to gray and remove noise points."""
    img = cv.imread(img_file)                  # Read image.

    img = cv.GaussianBlur(img, (3, 3),0)    # GaussianBlur
    img = cv.medianBlur(img, 3)            # Median Filtering
    img = cv.GaussianBlur(img, (3, 3),0)    # GaussianBlur
    img = cv.medianBlur(img, 3)            # Median Filtering

    return cv.cvtColor(img, cv.COLOR_BGR2GRAY)   # BGR to gray.

class plot_from_matplotlib:
    """
    Make pictures from matplotlib.

    # Example
    ```
    plot = plot_from_matplotlib("title")
    plot("Original",img)
    plot_from_matplotlib.show()
    ```
    """
    n:int=1
    def __init__(self,name:str) -> None:
        fig = plt.figure()
        fig.canvas.manager.window.setWindowTitle(name)

    def __call__(self,name:str,dst,cmap:str='gray',save:bool=1):     # Make pictures.
        n = self.n
        path = __file__.split("\\")[-2]
        plt.subplot(2,2,n), plt.imshow(dst,cmap), plt.title(name)
        plt.xticks([]), plt.yticks([])  # Do not show scale.
        if save:
cv.imwrite(f"./{path}/{name}.png",dst,[int(cv.IMWRITE_PNG_COMPRESSION),0])
```

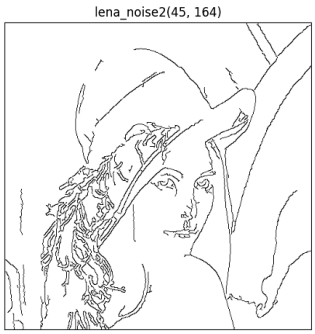
```
        self.n+=1

    @staticmethod
    def show():
        plt.show()

def canny(img:ndarray,min:int=0,max:int=255) -> ndarray:return
cv.bitwise_not(cv.Canny(img,min,max))

def canny_edge(img:ndarray) -> tuple[int,int]:
    """Test Canny upper and lower values."""
    cv.namedWindow("Canny")

    ### Create a track bar ### cv2.createTrackbar('Slider name', 'window name', min,
max, fn)
    cv.createTrackbar("minimum", "Canny", 0, 255, lambda _:_)
    cv.createTrackbar("maximum", "Canny", 0, 255, lambda _:_)
    cv.setTrackbarPos("minimum", "Canny", 0)
    cv.setTrackbarPos("maximum", "Canny", 255)

    while True:
        ### Get trackbar position. ### cv2.setTrackbarPos('Slider name', 'window
name', default)
        min= cv.getTrackbarPos('minimum','Canny')
        max= cv.getTrackbarPos('maximum','Canny')

        cv.imshow("Canny",canny(img,min,max))
        key = cv.waitKey(1)
        if  key == ord('q') or key == 27:
            break

    cv.destroyAllWindows()
    return min,max

### Convert bgr to gray and remove noise points. ###
lena = bgr2gray("lena.jpg")            # 60, 141
lena_n1 = bgr2gray("lena_noise.jpg",)   # 90, 169
lena_n2 = bgr2gray("lena_noise2.jpg")   # 45, 164
lena_n3 = bgr2gray("lena_noise3.jpg")   # 80, 161

### Test Canny upper and lower values. ###
print(canny_edge(lena))
```

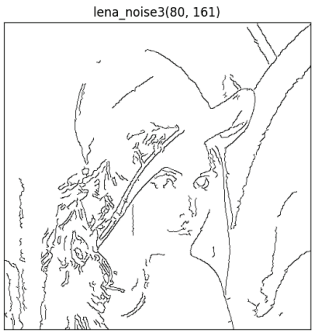
```
print(canny_edge(lena_n1))
print(canny_edge(lena_n2))
print(canny_edge(lena_n3))

### Plot the results. ###
plot = plot_from_matplotlib("Canny")
plot("lena(60, 141)",canny(lena,60, 141))
plot("lena_noise(90, 169)",canny(lena_n1,90, 169))
plot("lena_noise2(45, 164)",canny(lena_n2,45, 164))
plot("lena_noise3(80, 161)",canny(lena_n3,80, 161))
plot_from_matplotlib.show()
```

B、 Result map


lena(60, 141)


lena_noise(90, 169)


lena_noise2(45, 164)


lena_noise3(80, 161)

C、 Appendix (original picture)

Figure 1　lena(60, 141)

Figure 2　lena_noise(90, 169)

Figure 3　lena_noise2(45, 164)

Figure 4　lena_noise3(80, 161)