

Computer vision – HW5

A、程式碼

```
# WeiWen Wu 2023-10-18

from numpy import ndarray, array, float32
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread('lena_noise.jpg')

def filter2D(name:str, kernel:tuple[ndarray,ndarray], n:int):
    kernel_x, kernel_y = kernel

    dst = cv.filter2D(img, -1, kernel_x)
    dst = cv.filter2D(dst, -1, kernel_y)

    plt.subplot(n), plt.imshow(dst), plt.title(name)
    plt.xticks([]), plt.yticks([])

    cv.imwrite(f"./HW5/{name}.png", dst, [int(cv.IMWRITE_PNG_COMPRESSION), 0])

### Plot original image. ###
plt.subplot(231), plt.imshow(img), plt.title("Original")
plt.xticks([]), plt.yticks([])

# Define the kernel, using a N×N NumPy array, where N is odd.
roberts_operator = array([[1,0,0],[0,-1,0],[0,0,0]],float32), \
    array([[0,1,0],[-1,0,0],[0,0,0]],float32)
filter2D("Roberts operator", roberts_operator, 232)

prewitt_filters=array([[-1,0,1],[-1,0,1],[-1,0,1]],float32), \
    array([[-1,-1,-1],[0,0,0],[1,1,1]],float32)
filter2D("Prewitt filters", prewitt_filters, 233)

sobel_operator=array([[-1,0,1],[-2,0,2],[-1,0,1]],float32), \
    array([[-1,-2,-1],[0,0,0],[1,2,1]],float32)
filter2D("Sobel operator", sobel_operator, 234)

robinson_operator=array([[1,1,1],[1,-2,1],[-1,-1,-1]],float32), \
    array([[-1,1,1],[-1,-2,1],[-1,1,1]],float32)
filter2D("Robinson operator", robinson_operator, 235)
```

```

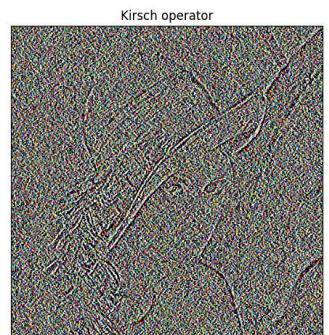
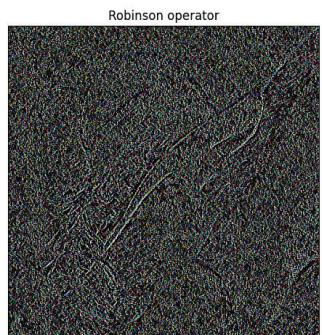
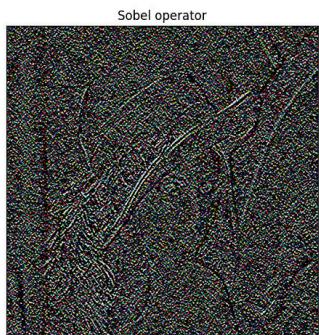
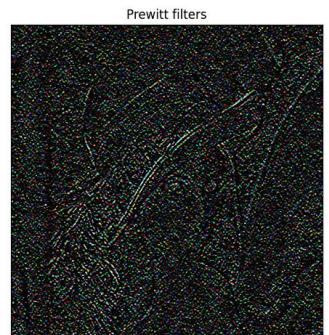
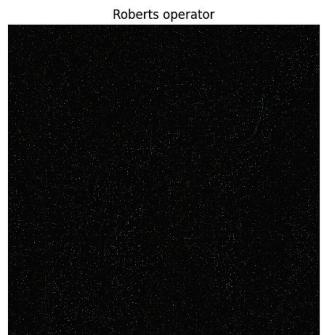
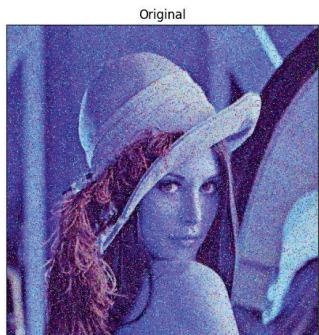
kirsch_operator=array([[3,3,3],[3,0,3],[-5,-5,-5]],float32), \
array([[-5,3,3],[-5,0,3],[-5,3,3]],float32)
filter2D("Kirsch operator",kirsch_operator,236)

### Show matplotlib. ###
plt.show()

```

B、成果圖

從下圖來看，使用基爾希運算符(Kirsch operator)效果較好。



C、附錄(原始圖片)

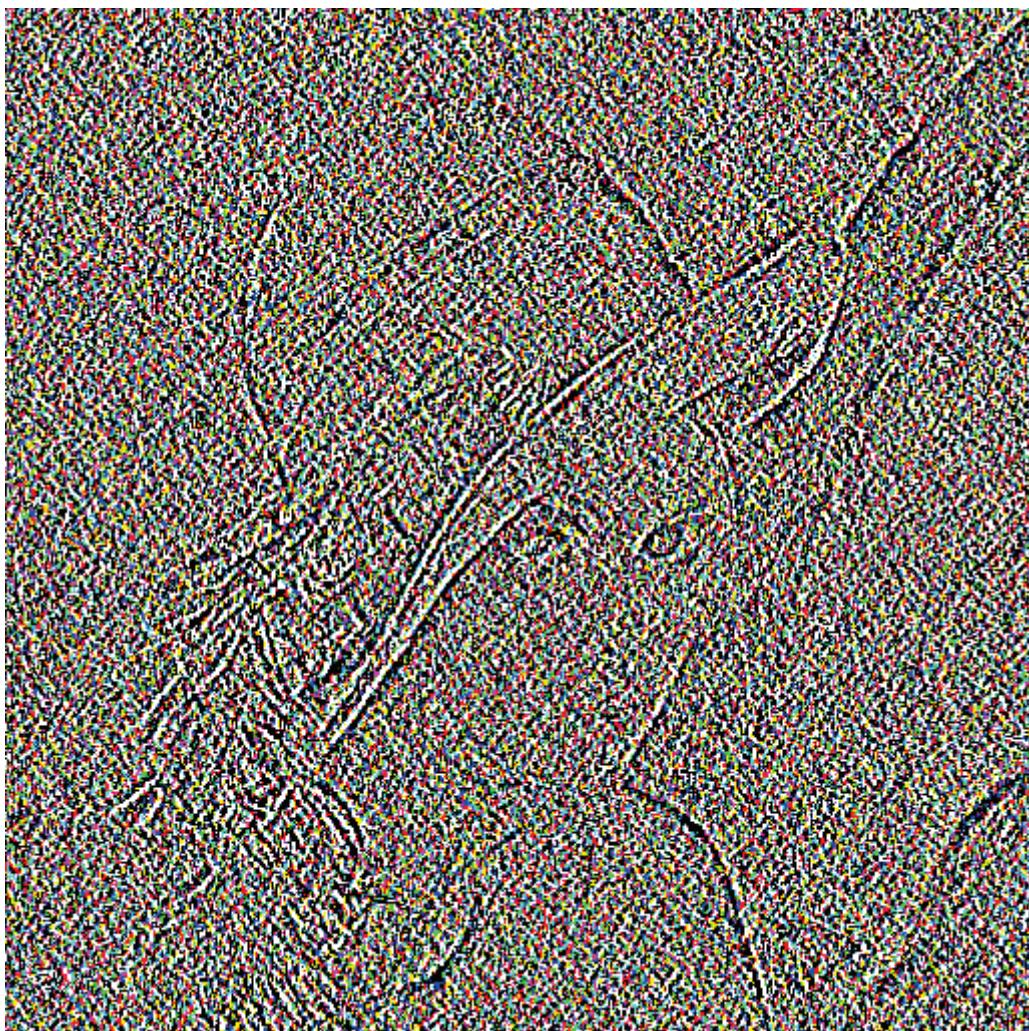


Figure 1 Kirsch operator

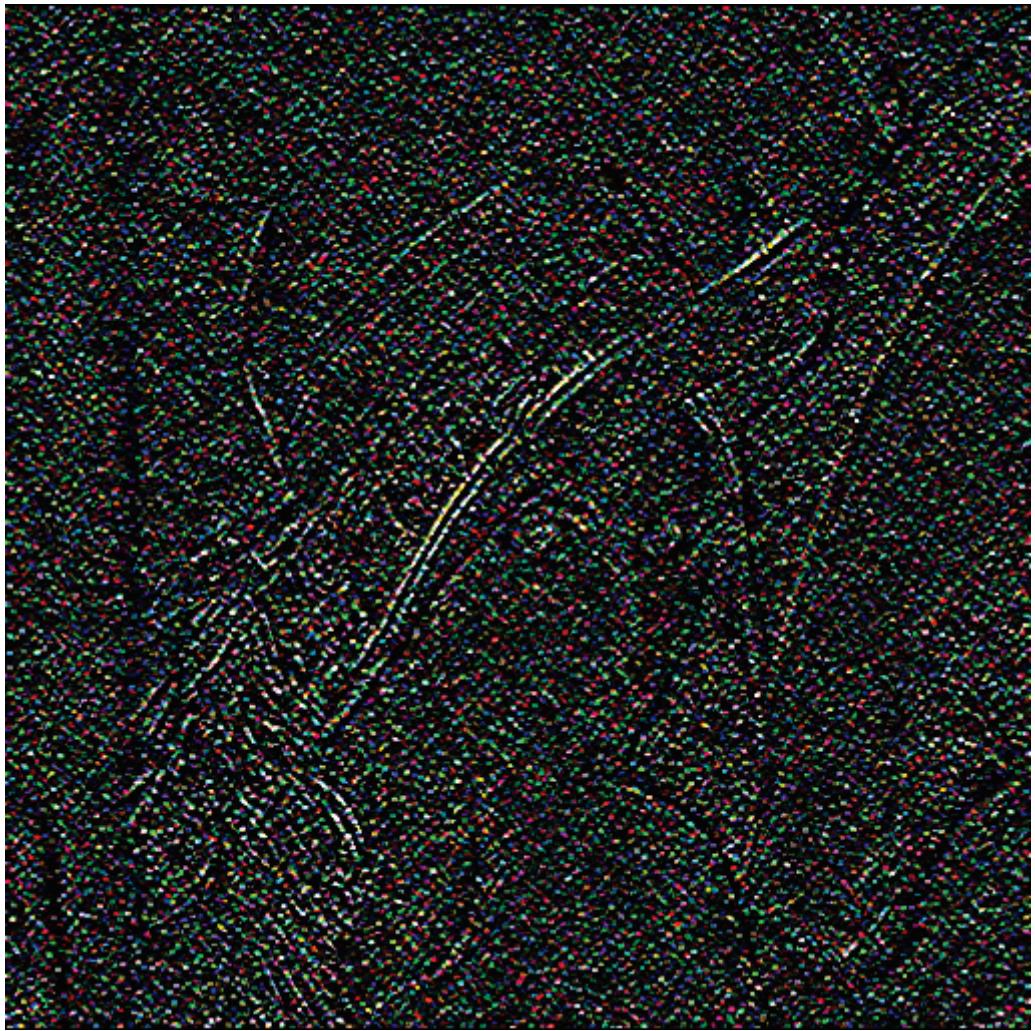


Figure 2 Prewitt filters

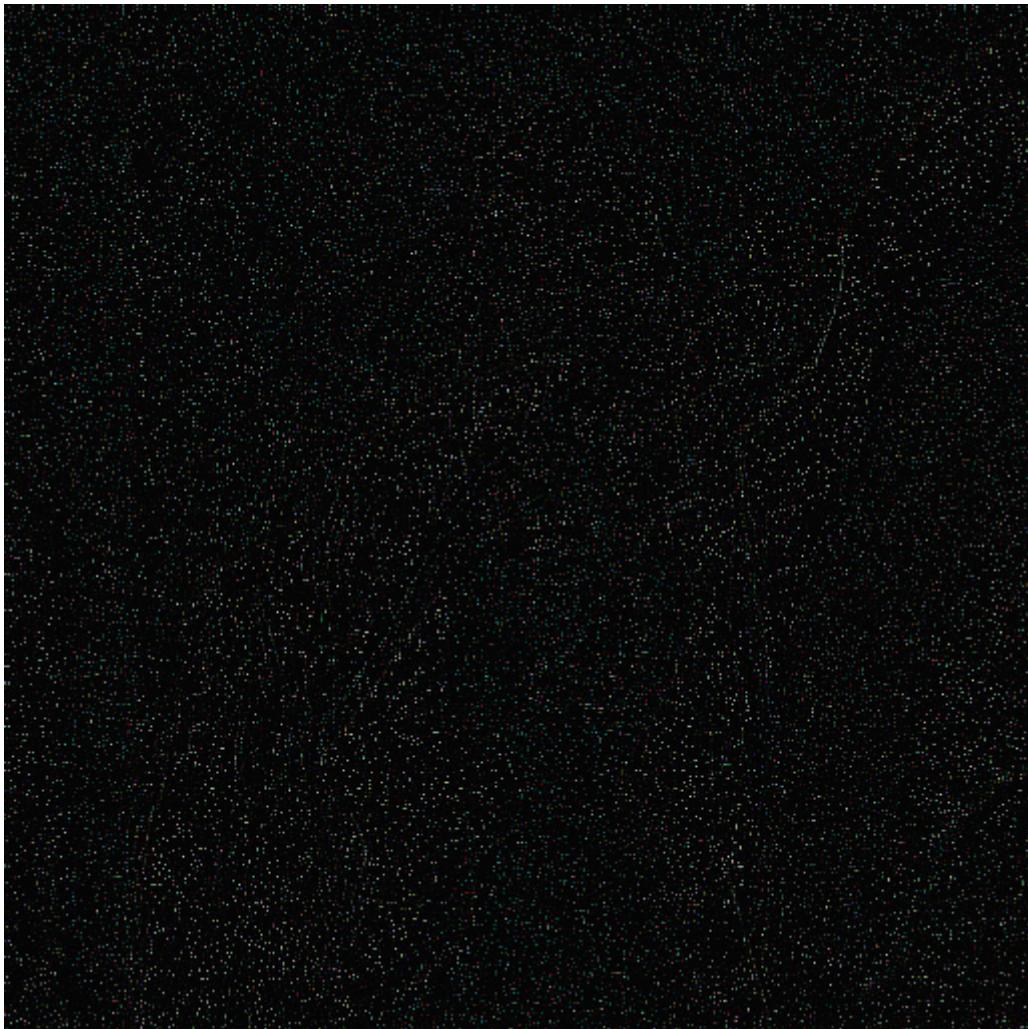


Figure 3 Roberts operator

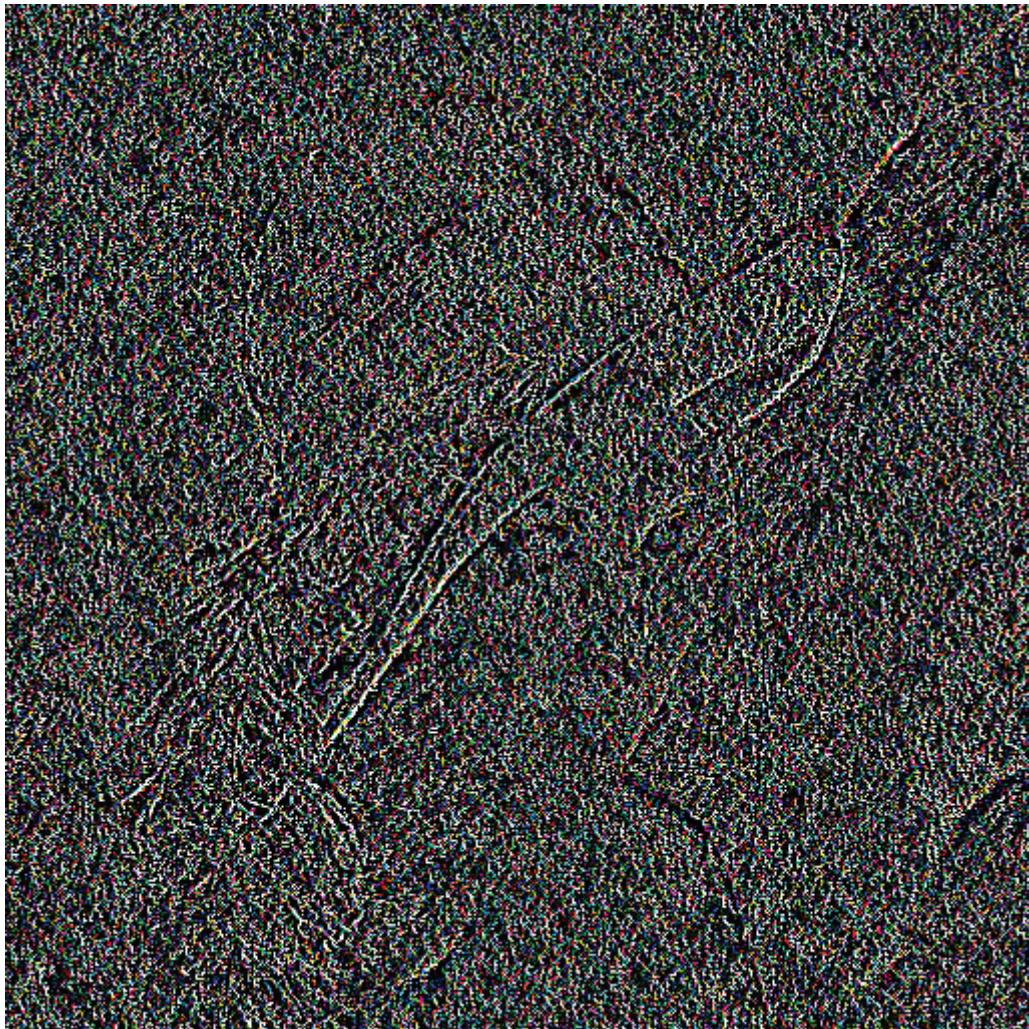


Figure 4 Robinson operator

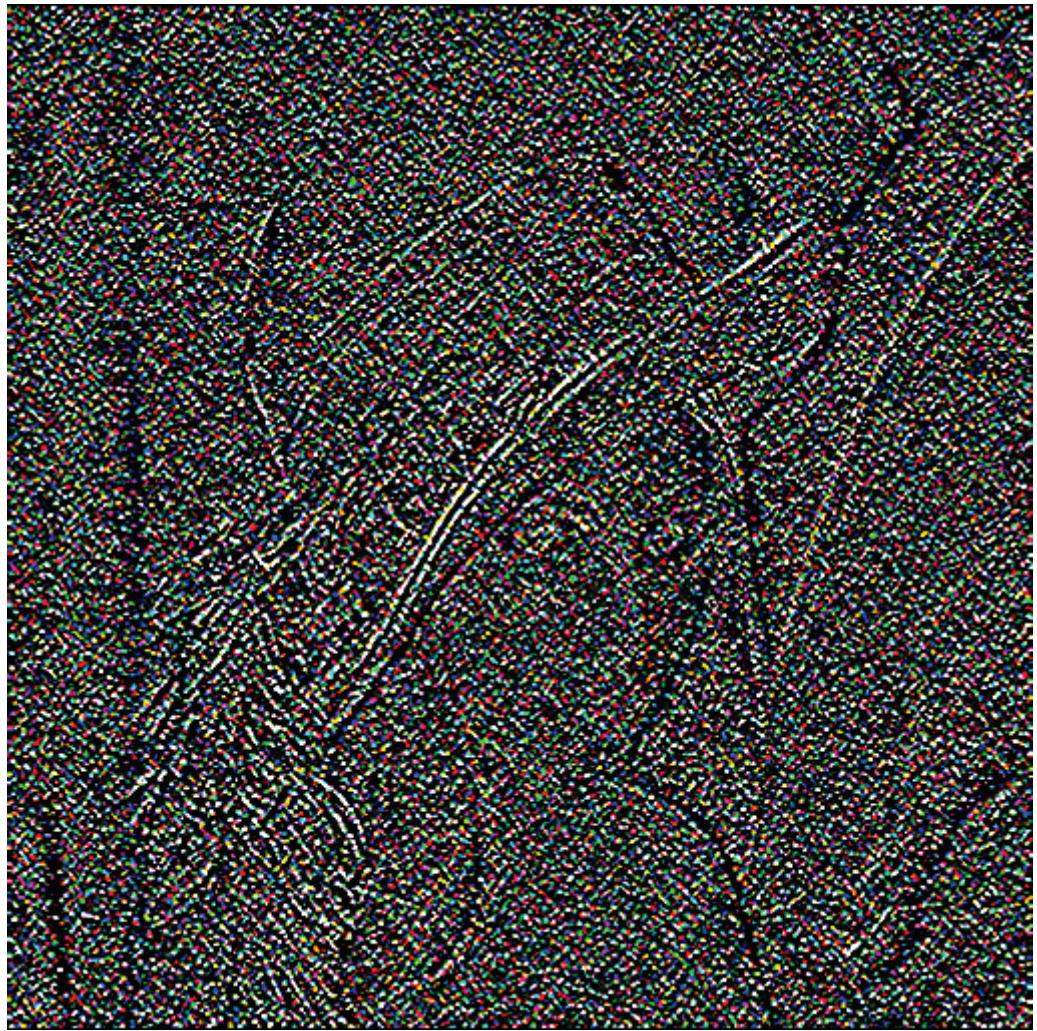


Figure 5 Sobel operator