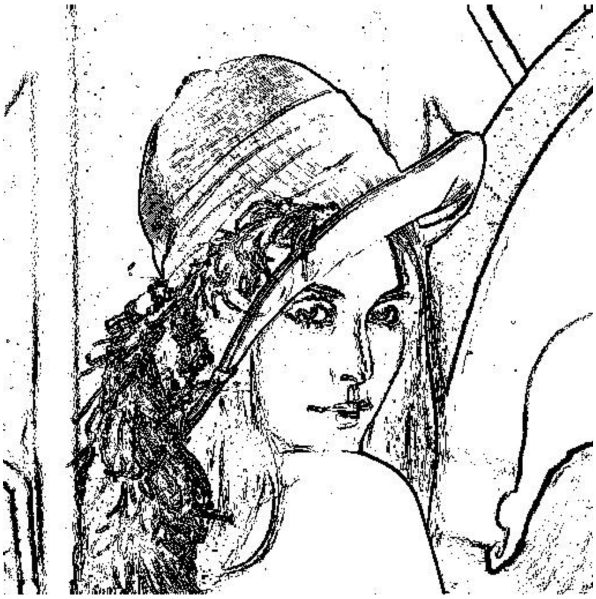
Computer Vision HW9

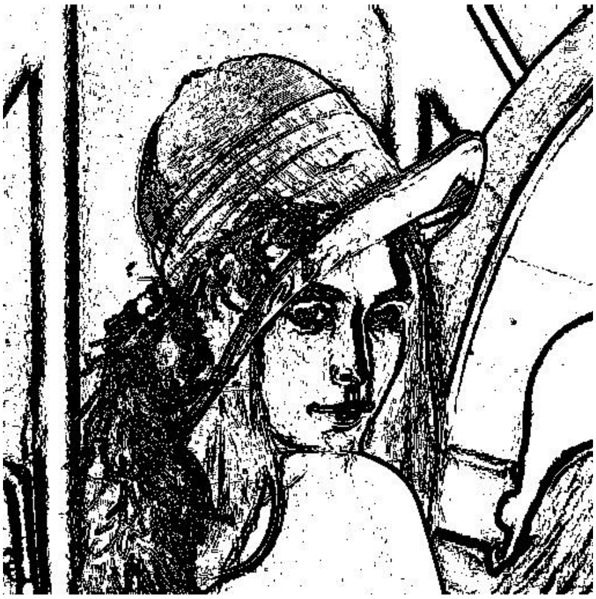
R08922156 黃劍韜

(a) Robert's Operator (Threshold 15)



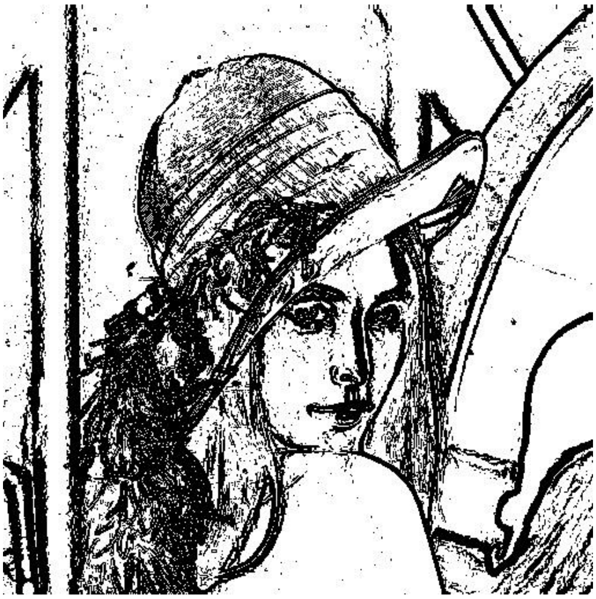
用[[-1, 0], [0, 1]]以及[[0, -1], [1, 0]]這兩個 Kernel 然後去做 convolution 然後計算平方和開根號作為 gradient magnitude。

(b) Prewitt's Edge Detector (Threshold 24)



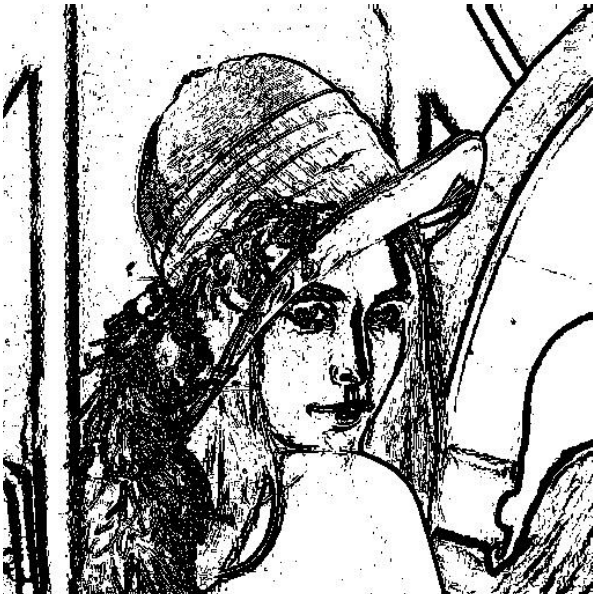
用[[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]以及[[-1, -1, -1], [0, 0, 0], [1, 1, 1]]這兩個 Kernel 然後去做 convolution 然後計算平方和開根號作為 gradient magnitude。

(c) Sobel's Edge Detector (Threshold 38)



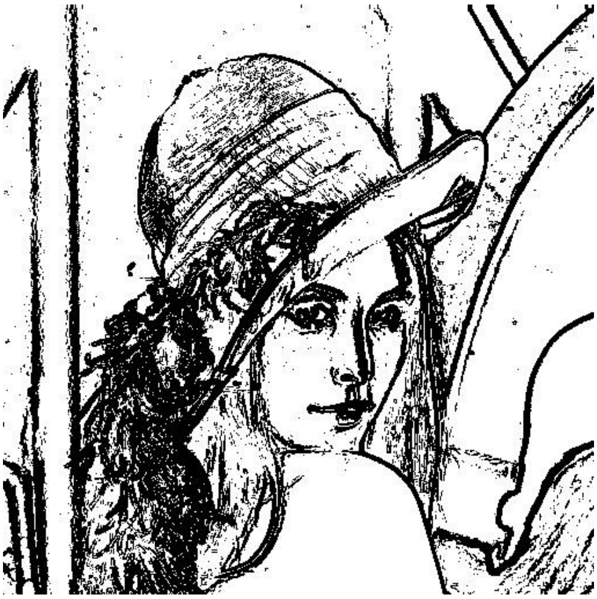
用[[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]以及[[-1, -2, -1], [0, 0, 0], [1, 2, 1]]這兩個 Kernel 然後去做 convolution 然後計算平方和開根號作為 gradient magnitude。

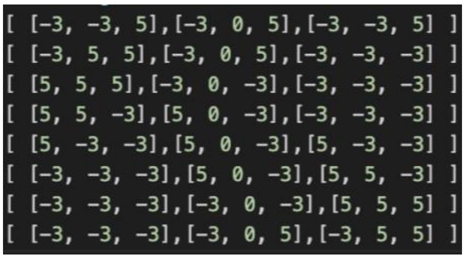
(d) Frei and Chen's Gradient Operator (Threshold 30)



用[[-1, -$\sqrt{2}$, -1], [0, 0, 0], [1, $\sqrt{2}$, 1]]以及[[-1, -2, -1], [0, 0, 0], [1, 2, 1]]這兩個 Kernel 然後去做 convolution 然後計算平方和開根號作為 gradient magnitude。

(e) Kirsch's Compass Operator (Threshold 135)

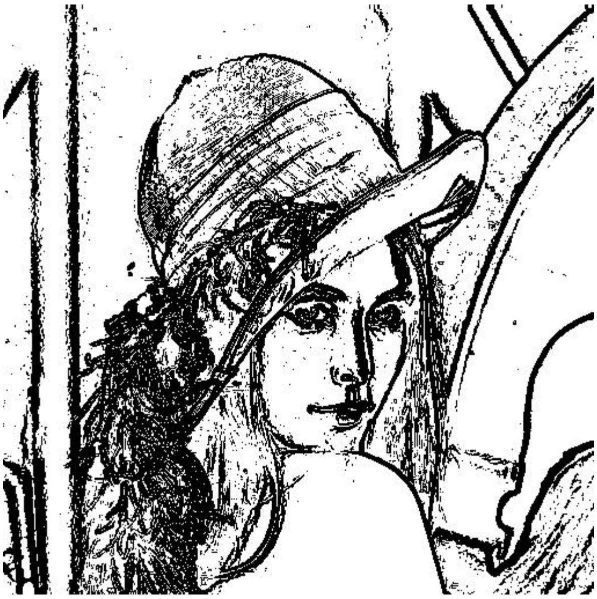


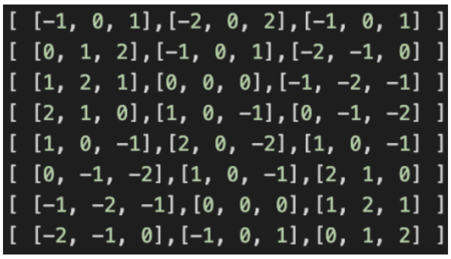


```
[ [-3, -3, 5],[-3, 0, 5],[-3, -3, 5] ]
[ [-3, 5, 5],[-3, 0, 5],[-3, -3, -3] ]
[ [5, 5, 5],[-3, 0, -3],[-3, -3, -3] ]
[ [5, 5, -3],[5, 0, -3],[-3, -3, -3] ]
[ [5, -3, -3],[5, 0, -3],[5, -3, -3] ]
[ [-3, -3, -3],[5, 0, -3],[5, 5, -3] ]
[ [-3, -3, -3],[-3, 0, -3],[5, 5, 5] ]
[ [-3, -3, -3],[-3, 0, 5],[-3, 5, 5] ]
```

用右上圖這七個 Kernel 然後去做 convolution 然後取其中的最大值作為 gradient magnitude 然後去做 edge operator。

(f) Robinson's Compass Operator (Threshold 43)

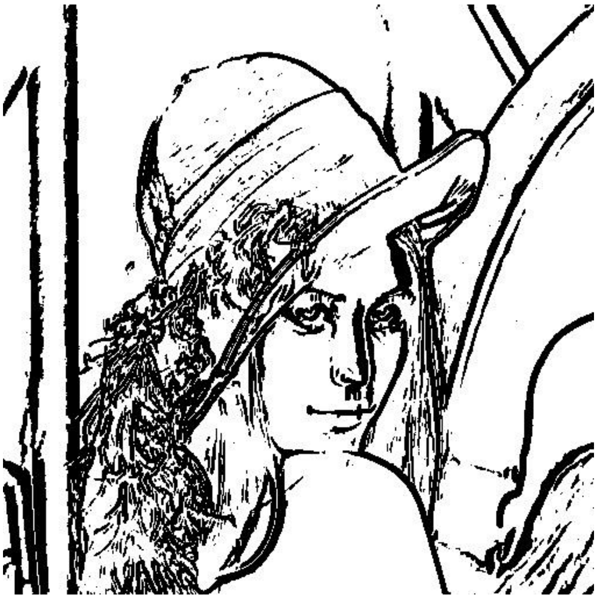




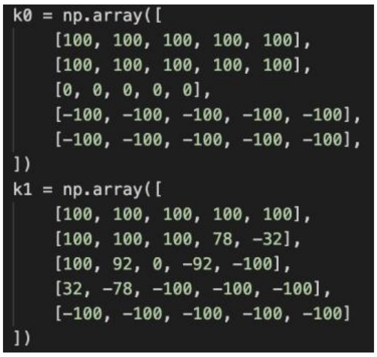
```
[ [-1, 0, 1],[-2, 0, 2],[-1, 0, 1] ]
[ [0, 1, 2],[-1, 0, 1],[-2, -1, 0] ]
[ [1, 2, 1],[0, 0, 0],[-1, -2, -1] ]
[ [2, 1, 0],[1, 0, -1],[0, -1, -2] ]
[ [1, 0, -1],[2, 0, -2],[1, 0, -1] ]
[ [0, -1, -2],[1, 0, -1],[2, 1, 0] ]
[ [-1, -2, -1],[0, 0, 0],[1, 2, 1] ]
[ [-2, -1, 0],[-1, 0, 1],[0, 1, 2] ]
```

用右上圖這七個 Kernel 然後去做 convolution 然後取其中的最大值作為 gradient magnitude 然後去做 edge operator。

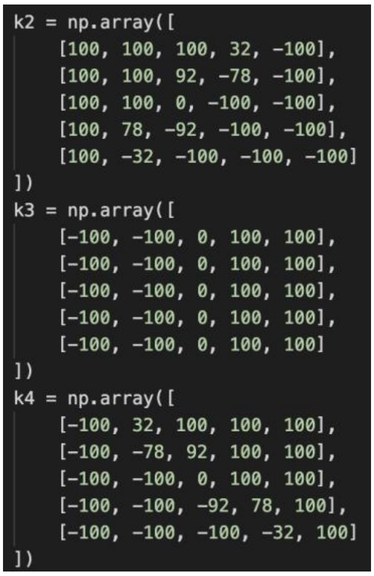
(g) Robinson's Compass Operator (Threshold 12500)



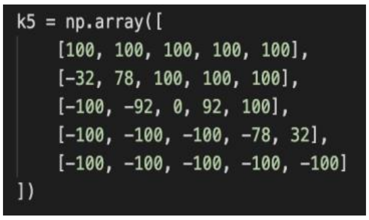
用右圖這五個 Kernel 然後去做 convolution 然後取其中的最大值作為 gradient magnitude 然後去做 edge detection。



```
k0 = np.array([
    [100, 100, 100, 100, 100],
    [100, 100, 100, 100, 100],
    [0, 0, 0, 0, 0],
    [-100, -100, -100, -100, -100],
    [-100, -100, -100, -100, -100],
])
k1 = np.array([
    [100, 100, 100, 100, 100],
    [100, 100, 100, 78, -32],
    [100, 92, 0, -92, -100],
    [32, -78, -100, -100, -100],
    [-100, -100, -100, -100, -100]
])
```



```
k2 = np.array([
    [100, 100, 100, 32, -100],
    [100, 100, 92, -78, -100],
    [100, 100, 0, -100, -100],
    [100, 78, -92, -100, -100],
    [100, -32, -100, -100, -100]
])
k3 = np.array([
    [-100, -100, 0, 100, 100],
    [-100, -100, 0, 100, 100],
    [-100, -100, 0, 100, 100],
    [-100, -100, 0, 100, 100],
    [-100, -100, 0, 100, 100]
])
k4 = np.array([
    [-100, 32, 100, 100, 100],
    [-100, -78, 92, 100, 100],
    [-100, -100, 0, 100, 100],
    [-100, -100, -92, 78, 100],
    [-100, -100, -100, -32, 100]
])
```



```
k5 = np.array([
    [100, 100, 100, 100, 100],
    [-32, 78, 100, 100, 100],
    [-100, -92, 0, 92, 100],
    [-100, -100, -100, -78, 32],
    [-100, -100, -100, -100, -100]
])
```