電腦視覺 Homework 10 – Zero Crossing Edge Detection

資工碩二 r08922a04 林承德

Language : python 3.7
Library : Numpy, opencv(cv2)
Execution way: python3 hw10.py
(please put lena.bmp at the same directory with hw10.py)

Description:

幾種不同的處理方式的核心演算法幾乎相通，主要的差異在每種作法使用的mask或padding都不同，而他們主要的作法如下：將輸入的原圖做padding(Laplacian與minimum variance laplacian的padding size=1, laplacian of gaussian and difference of gaussian的padding size = 5)，接著每個種作法使用其各自對應的mask，算出gradient magnitude，並將結果與給定的threshold比對，並檢查zero crossing，找出edge

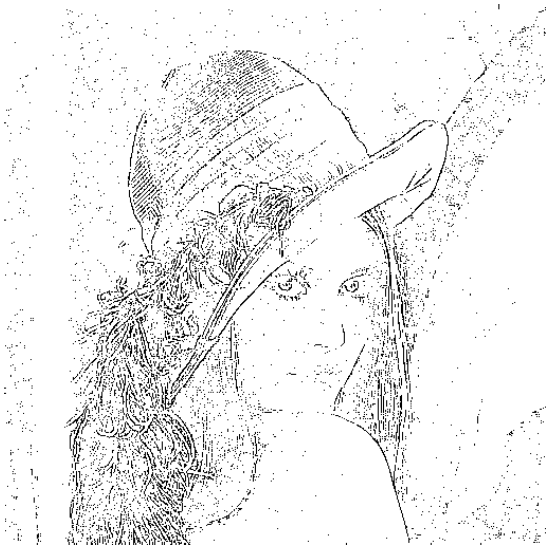
Result:

(A) Laplacian Mask 1
threshold = 15

kernel

```
# Laplacian_mask1
l_mask1 = np.array([[0, 1, 0],
                    [1, -4, 1],
                    [0, 1, 0]])
```

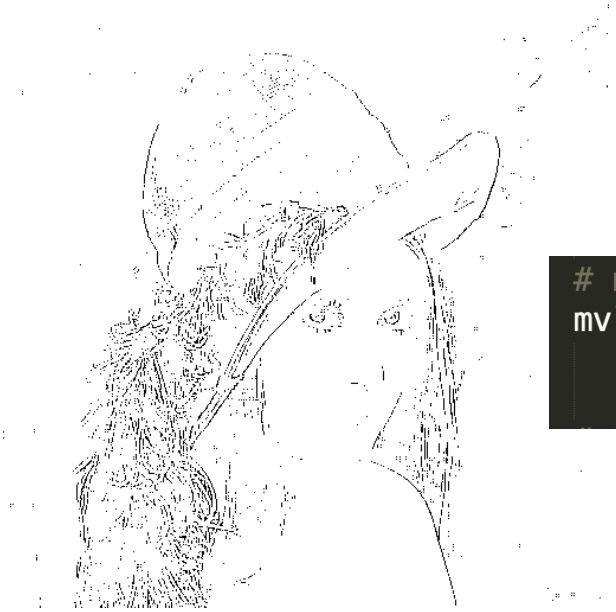
(B) Laplacian Mask 2
threshold = 15



kernel

```
                                [0, 1, 0]])
# Laplacian_mask2
l_mask2 = (1/3) * np.array([[1, 1, 1],
                            [1, -8, 1],
                            [1, 1, 1]])
```

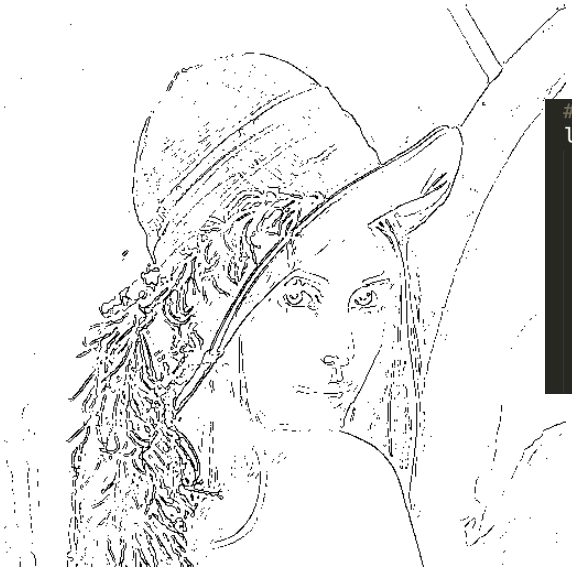
(C) Minimum Variance Laplacian
threshold = 20



kernel

```
# minimum varience laplacian mask
mvl_mask = (1/3) * np.array([[2, -1, 2],
                             [-1, -4, -1],
                             [2, -1, 2]])
```

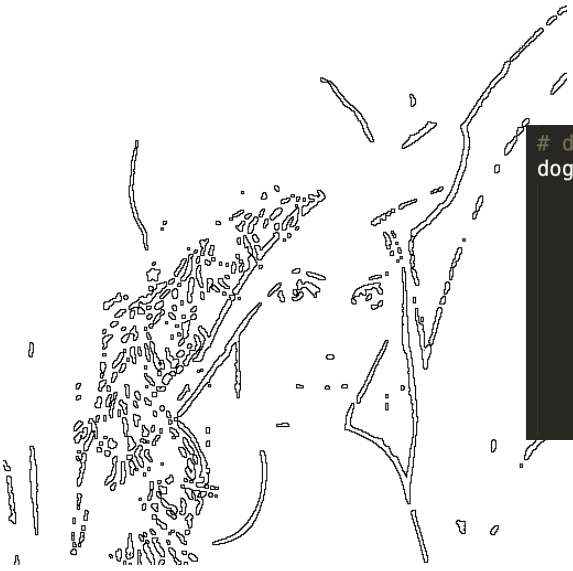
(D) Laplacian of Gaussian
threshold = 3000



kernel

```
# laplacian of guassian mask
log_mask = np.array([[0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0],
                     [0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],
                     [0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],
                     [-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],
                     [-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],
                     [-2, -9, -23, -1, 103, 178, 103, -1, -23, -9, -2],
                     [-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],
                     [-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],
                     [0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],
                     [0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],
                     [0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0]])
```

(E) Difference of Gaussian
threshold = 1



kernel

```
# difference of guassian mask
dog_mask = np.array([[-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1],
                     [-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],
                     [-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],
                     [-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],
                     [-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],
                     [-8, -13, -17, 15, 160, 283, 160, 15, -17, -13, -8],
                     [-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],
                     [-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],
                     [-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],
                     [-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],
                     [-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1]])
```