電腦視覺 HW10

R10922176 陳冠穎

說明：此次作業都是使用自己實作的 correlation 與 zero-crossing edge detection 函式對每種 mask 作用。

```python
def correlation(img, mask, threshold):
    half_mask_size = int(len(mask[0]) / 2)
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, half_mask_size, half_mask_size, half_mask_size,
                               half_mask_size, cv2.BORDER_REPLICATE)

    for r in range(half_mask_size, img.shape[0] + half_mask_size):
        for c in range(half_mask_size, img.shape[1] + half_mask_size):
            if(apply_mask(eximg, r, c, mask) >= threshold):
                new_img[r-half_mask_size][c-half_mask_size] = 1
            elif(apply_mask(eximg, r, c, mask) > -threshold):
                new_img[r-half_mask_size][c-half_mask_size] = 0
            else:
                new_img[r-half_mask_size][c-half_mask_size] = -1

    return new_img
```

```python
def zero_crossing(img):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)

    offset = 1
    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            is_cross = False
            if(eximg[r][c] == 1):
                for i in range(-offset, offset + 1):
                    for j in range(-offset, offset + 1):
                        if(i != 0 and j != 0):
                            if(eximg[r+i][c+j] == -1):
                                is_cross = True
                                break
                    if is_cross:
                        break

            new_img[r-1][c-1] = 0 if is_cross else 255
    return new_img
```

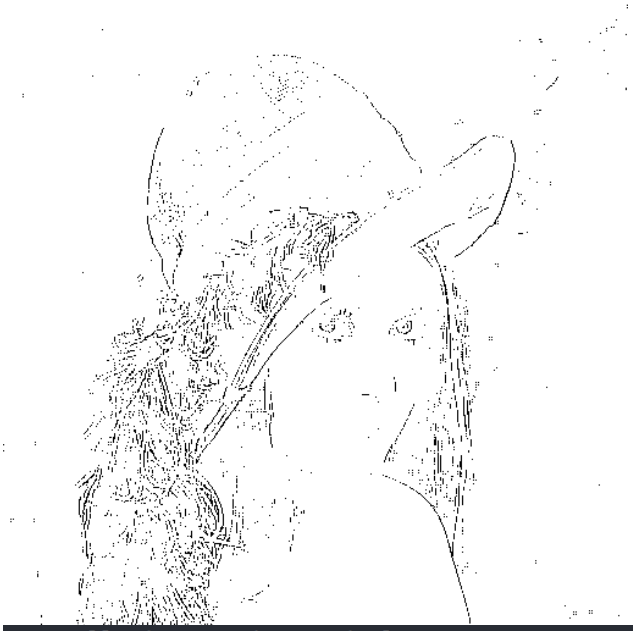
(a) Laplace Mask1 (0, 1, 0, 1, -4, 1, 0, 1, 0): 15

```
m1 = [[0, 1, 0],
      [1, -4, 1],
      [0, 1, 0]]
laplacian1 = zero_crossing(correlation(img, m1, 15))
cv2.imwrite(output_file_path + 'laplacian1' + '.bmp', laplacian1)
```

(b) Laplace Mask2 (1, 1, 1, 1, -8, 1, 1, 1, 1): 15

```
m2 = [[1/3, 1/3, 1/3],
      [1/3, -8/3, 1/3],
      [1/3, 1/3, 1/3]]
laplacian2 = zero_crossing(correlation(img, m2, 15))
cv2.imwrite(output_file_path + 'laplacian2' + '.bmp', laplacian2)
```

(c) Minimum variance Laplacian: 20



```
m3 = [[2 / 3, -1 / 3, 2 / 3],
      [-1 / 3, -4 / 3, -1 / 3],
      [2 / 3, -1 / 3, 2 / 3]]
min_laplacian = zero_crossing(correlation(img, m3, 20))
cv2.imwrite(output_file_path + 'min_laplacian' + '.bmp', min_laplacian)
```

(d) Laplace of Gaussian: 3000

```
m4 = [[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]]
laplace_of_g = zero_crossing(correlation(img, m4, 3000))
cv2.imwrite(output_file_path + 'laplace_of_g' + '.bmp', laplace_of_g)
```

(e) Difference of Gaussian: 1

```
m5 = [[-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]]
difference_of_g = zero_crossing(correlation(img, m5, 1))
cv2.imwrite(output_file_path + 'difference_of_g' + '.bmp', difference_of_g)
```