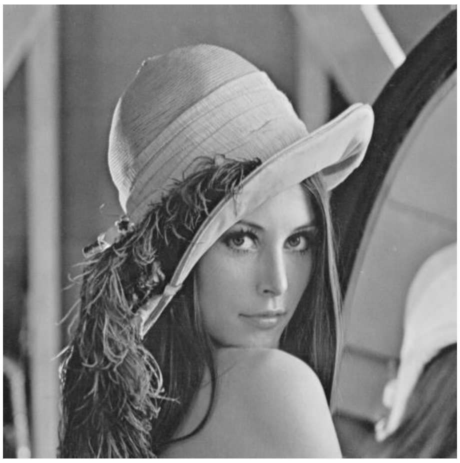
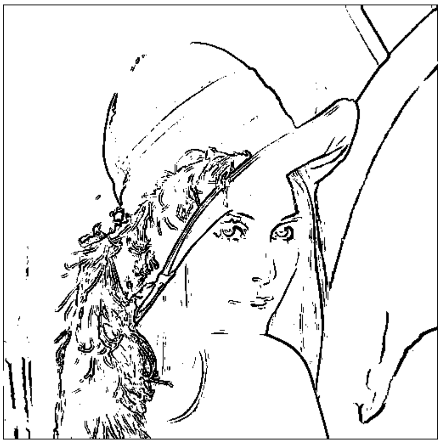
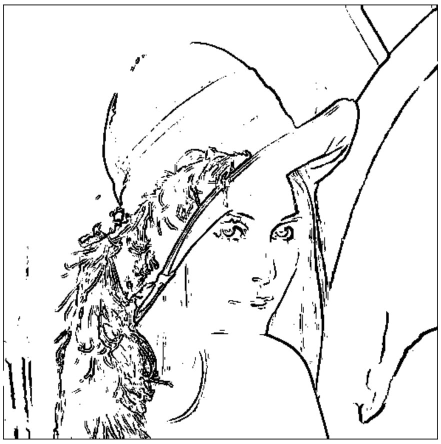
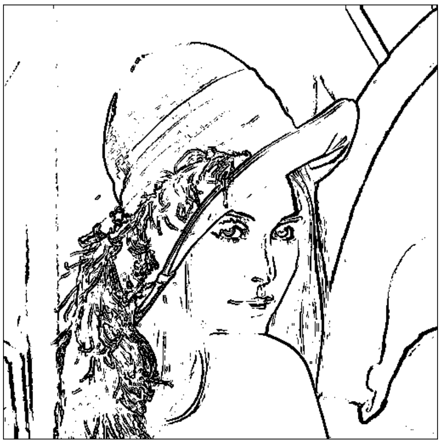
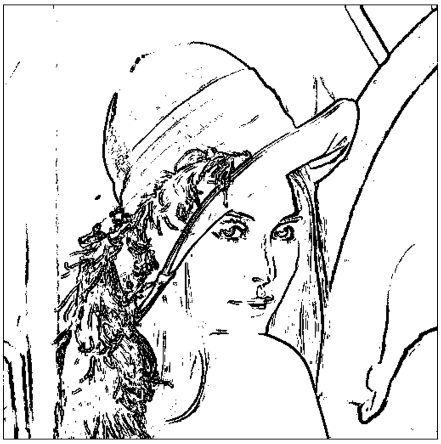
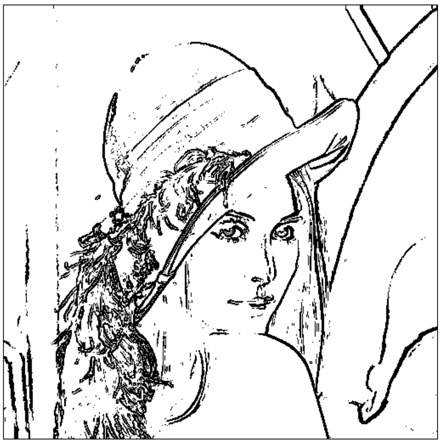
Homework 10 R08921005 黃國郡

Answer

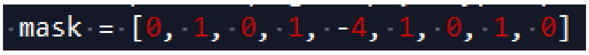
 Lena.bmp	 Laplacian mask 1 Threshold =20
 Laplacian2 Thresholds:20	 Minimum variance Laplacian Thresholds:15
 Laplacian of Gaussian Thresholds:3000	 Difference of Gaussian Thresholds:1

Homework 10 R08921005 黃國郡

Algorithm

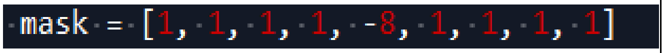
Each kernel of function

(a) Laplacian mask 1, Threshold =20



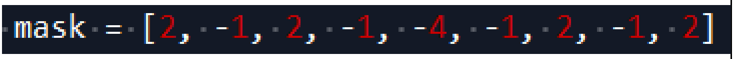
```
mask = [0, 1, 0, 1, -4, 1, 0, 1, 0]
```

(b) Laplacian2, Thresholds:20



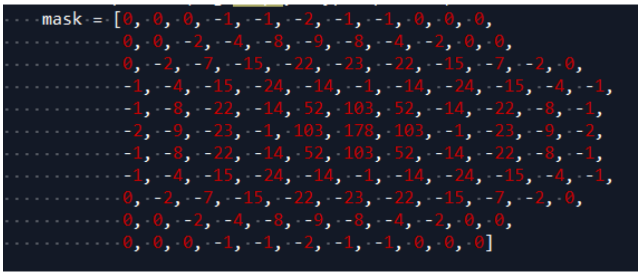
```
mask = [1, 1, 1, 1, -8, 1, 1, 1, 1]
```

(c) Minimum variance Laplacian, Thresholds:15



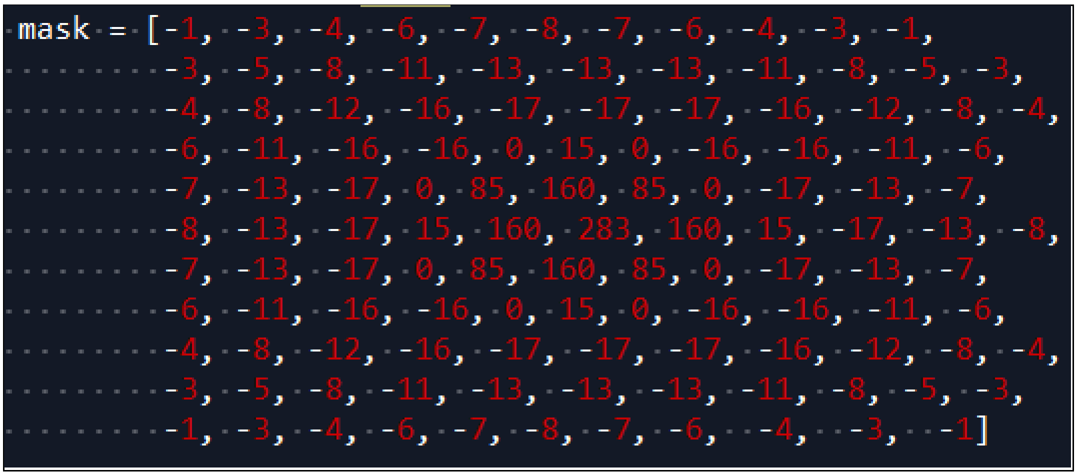
```
mask = [2, -1, 2, -1, -4, -1, 2, -1, 2]
```

(d) Laplacian of Gaussian, Thresholds:3000



```
    mask = [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 Thresholds:1



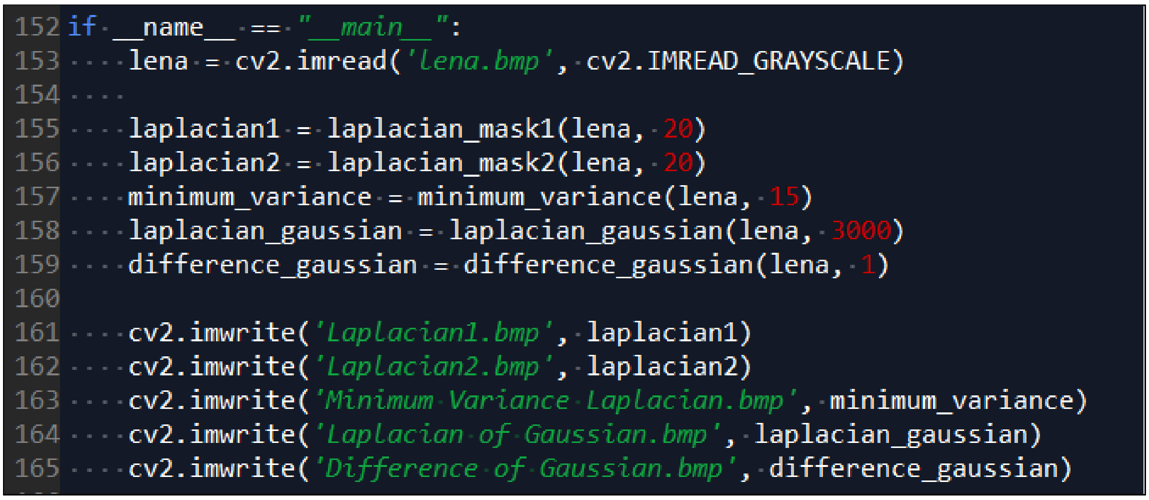
```
mask = [-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1,
        -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3,
        -4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4,
        -6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6,
        -7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7,
        -8, -13, -17, 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]
```

Homework 10　　R08921005 黃國郡

Code description

本次作業使用 python3.7，IDE 使用 Spyder

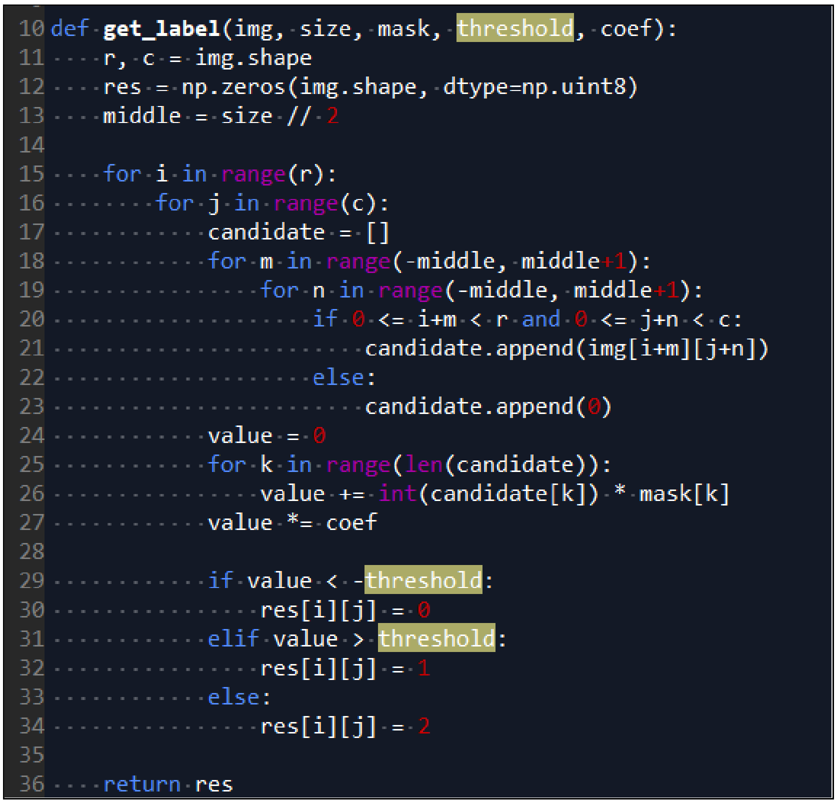
(a) Main function:



```python
if __name__ == "__main__":
    lena = cv2.imread('lena.bmp', cv2.IMREAD_GRAYSCALE)
    
    laplacian1 = laplacian_mask1(lena, 20)
    laplacian2 = laplacian_mask2(lena, 20)
    minimum_variance = minimum_variance(lena, 15)
    laplacian_gaussian = laplacian_gaussian(lena, 3000)
    difference_gaussian = difference_gaussian(lena, 1)

    cv2.imwrite('Laplacian1.bmp', laplacian1)
    cv2.imwrite('Laplacian2.bmp', laplacian2)
    cv2.imwrite('Minimum Variance Laplacian.bmp', minimum_variance)
    cv2.imwrite('Laplacian of Gaussian.bmp', laplacian_gaussian)
    cv2.imwrite('Difference of Gaussian.bmp', difference_gaussian)
```

沒什麼特別，輸入格別 function，再寫入 bmp 檔

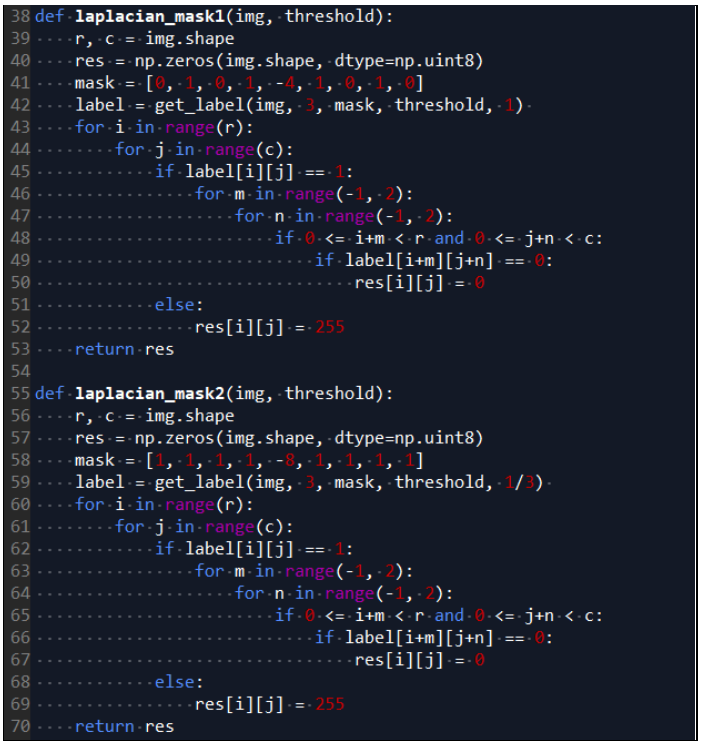
(b) get_label:



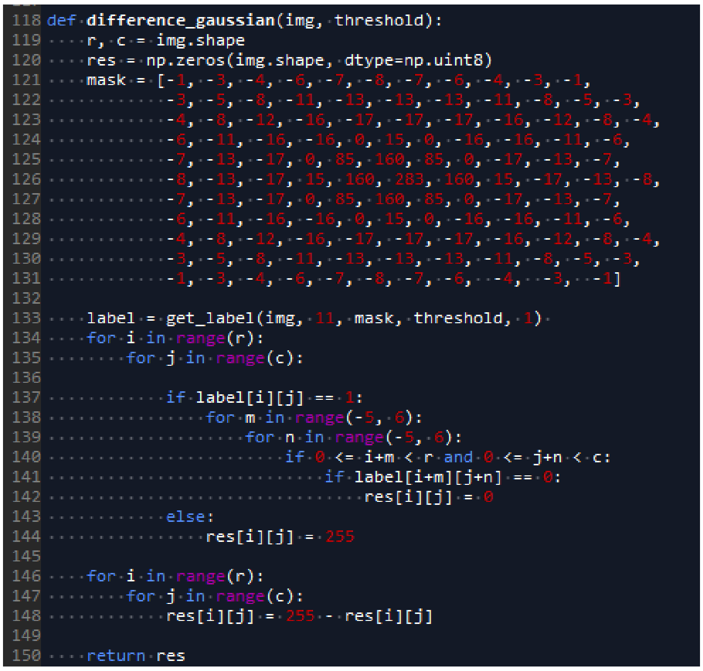
```python
def get_label(img, size, mask, threshold, coef):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    middle = size // 2

    for i in range(r):
        for j in range(c):
            candidate = []
            for m in range(-middle, middle+1):
                for n in range(-middle, middle+1):
                    if 0 <= i+m < r and 0 <= j+n < c:
                        candidate.append(img[i+m][j+n])
                    else:
                        candidate.append(0)
            value = 0
            for k in range(len(candidate)):
                value += int(candidate[k]) * mask[k]
            value *= coef

            if value < -threshold:
                res[i][j] = 0
            elif value > threshold:
                res[i][j] = 1
            else:
                res[i][j] = 2

    return res
```

主要為判斷 threshold 以便往後做判定是否符合 Zero Crossing Edge 條件

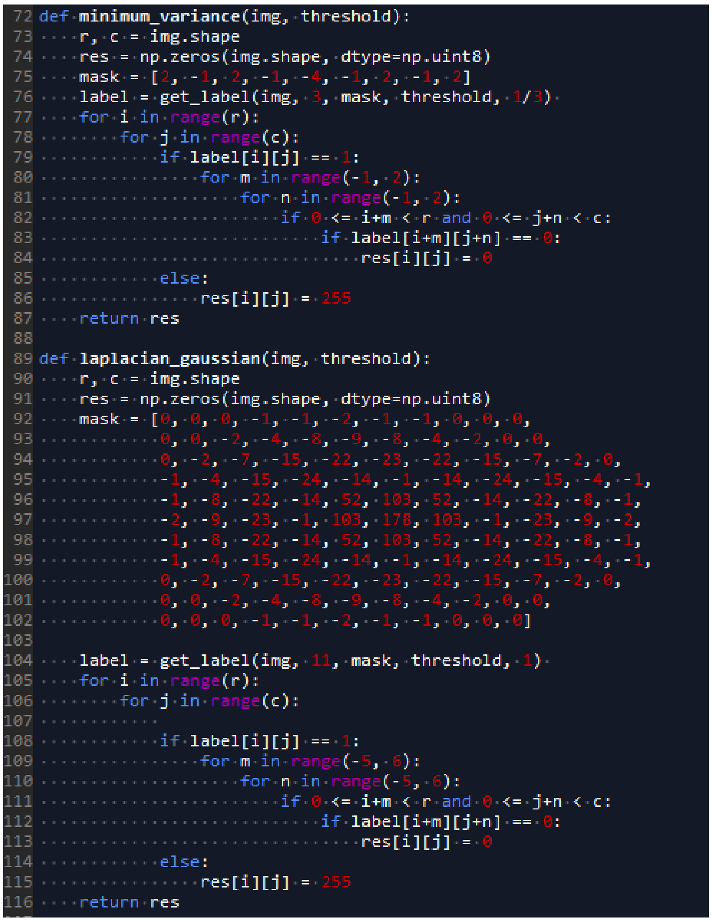
Homework 10　　R08921005 黃國郡



```python
def laplacian_mask1(img, threshold):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    mask = [0, 1, 0, 1, -4, 1, 0, 1, 0]
    label = get_label(img, 3, mask, threshold, 1)
    for i in range(r):
        for j in range(c):
            if label[i][j] == 1:
                for m in range(-1, 2):
                    for n in range(-1, 2):
                        if 0 <= i+m < r and 0 <= j+n < c:
                            if label[i+m][j+n] == 0:
                                res[i][j] = 0
            else:
                res[i][j] = 255
    return res

def laplacian_mask2(img, threshold):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    mask = [1, 1, 1, 1, -8, 1, 1, 1, 1]
    label = get_label(img, 3, mask, threshold, 1/3)
    for i in range(r):
        for j in range(c):
            if label[i][j] == 1:
                for m in range(-1, 2):
                    for n in range(-1, 2):
                        if 0 <= i+m < r and 0 <= j+n < c:
                            if label[i+m][j+n] == 0:
                                res[i][j] = 0
            else:
                res[i][j] = 255
    return res
```



```python
def difference_gaussian(img, threshold):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    mask = [-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]

    label = get_label(img, 11, mask, threshold, 1)
    for i in range(r):
        for j in range(c):

            if label[i][j] == 1:
                for m in range(-5, 6):
                    for n in range(-5, 6):
                        if 0 <= i+m < r and 0 <= j+n < c:
                            if label[i+m][j+n] == 0:
                                res[i][j] = 0
            else:
                res[i][j] = 255

    for i in range(r):
        for j in range(c):
            res[i][j] = 255 - res[i][j]

    return res
```



```python
def minimum_variance(img, threshold):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    mask = [2, -1, 2, -1, -4, -1, 2, -1, 2]
    label = get_label(img, 3, mask, threshold, 1/3)
    for i in range(r):
        for j in range(c):
            if label[i][j] == 1:
                for m in range(-1, 2):
                    for n in range(-1, 2):
                        if 0 <= i+m < r and 0 <= j+n < c:
                            if label[i+m][j+n] == 0:
                                res[i][j] = 0
            else:
                res[i][j] = 255
    return res

def laplacian_gaussian(img, threshold):
    r, c = img.shape
    res = np.zeros(img.shape, dtype=np.uint8)
    mask = [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]

    label = get_label(img, 11, mask, threshold, 1)
    for i in range(r):
        for j in range(c):
            
            if label[i][j] == 1:
                for m in range(-5, 6):
                    for n in range(-5, 6):
                        if 0 <= i+m < r and 0 <= j+n < c:
                            if label[i+m][j+n] == 0:
                                res[i][j] = 0
            else:
                res[i][j] = 255
    return res
```

以上為五個 mask 來判斷不同的梯度，也沒特別技巧，都依照上課 PPT 上去做