

## 電腦視覺 HW4

R10922176 陳冠穎

(a) Dilation



說明:

```
def dilation(img, mask):  
  
    new_img = np.zeros((img.shape[0], img.shape[1]))  
  
    for row in range(img.shape[0]):  
        for col in range(img.shape[1]):  
            maskUnion(new_img, img, row, col, mask)  
  
    return new_img
```

```
def maskUnion(new_img, img, r, c, mask):  
    if (img[r][c] != 0):  
        for i in range(-2, 3):  
            for j in range(-2, 3):  
                if (r + i >= 0 and r + i < img.shape[0] and c + j >= 0 and c + j < img.shape[1]):  
                    new_img[r + i][c + j] = new_img[r + i][c + j] or mask[2 + i][2 + j]
```

將 mask 滑過整張影像，比較該 pixel 與 mask 的中心，如果一樣

都是 1，那就把 mask 中為 1 的 pixel 照位置複製上去。

(b) Erosion



說明：

```
def maskMatch(img, r, c, mask):  
  
    for i in range(-2, 3):  
        for j in range(-2, 3):  
            if (r + i >= 0 and r + i < img.shape[0] and c + j >= 0 and c + j < img.shape[1]):  
                if mask[2 + i][2 + j] == 1 and img[r + i][c + j] == 0:  
                    return False  
  
            return True  
  
def erosion(img, mask):  
  
    new_img = np.zeros((img.shape[0], img.shape[1]))  
  
    for row in range(img.shape[0]):  
        for col in range(img.shape[1]):  
            if maskMatch(img, row, col, mask):  
                new_img[row][col] = 1  
  
    return new_img
```

將 mask 滑過整張圖片，如果經過比對 mask 為 1 的點，在原影像上面也都為 1，則 mask 的中心點在新影像上也設定為 1。

(c) Opening



說明:

```
def opening(img, mask):  
    return dilation(erosion(img, mask), mask)
```

使用前面寫好的 function，先做 erosion 在做 dilation。

(d) Closing

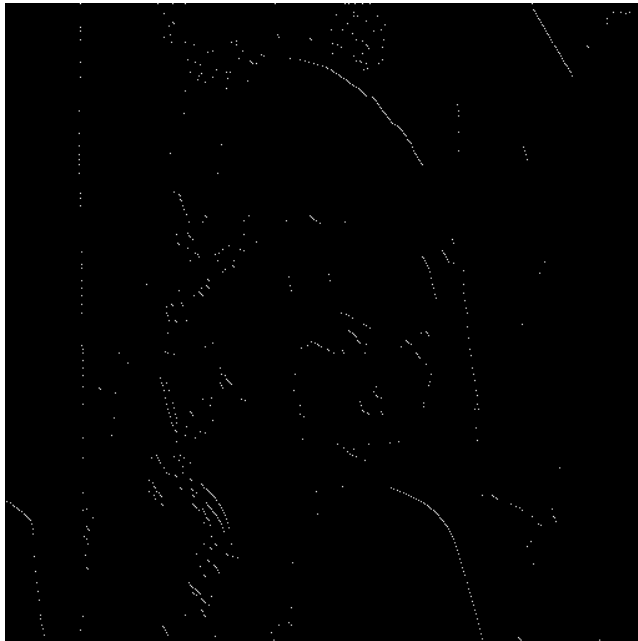


說明:

```
def closing(img, mask):  
    return erosion(dilation(img, mask), mask)
```

使用前面寫好的 function，先做 dilation 在做 erosion.

(e) Hit-and-miss transform



說明:

```
def hitAndMiss(img, j, k):  
  
    img_ero_j = erosion(img, j)  
    imgc_ero_k = erosion(complement(img), k)  
  
    new_img = np.zeros((img.shape[0], img.shape[1]))  
  
    for i in range(img.shape[0]):  
        for j in range(img.shape[1]):  
            if(img_ero_j[i][j] == 1 and imgc_ero_k[i][j] == 1):  
                new_img[i][j] = 1  
  
    return new_img
```

自己設定 j 跟 k 的 mask，然後原影像先做跟 j 做 erosion，原影像取補集(自己寫的 complement function)後跟 k 也做 erosion，在把他們取交集。