

電腦視覺 (一) Homework5

R11521701 程懷恩

1. Dilation



Code fragment:

```
def dilation(img,kernel):
    arr = np.zeros(img.shape, np.uint8)
    for i in range(0, img.shape[0]):
        for j in range(0, img.shape[1]):
            if img[i][j]!=0:
                max = 0
                for ite in kernel:
                    x, y = ite
                    if x+i>=0 and x+i<img.shape[0] and y+j>=0 and y+j<img.shape[1]:
                        if img[x+i][y+j]>max: max=img[x+i][y+j]
                arr[i][j]=max
    return arr
```

根據課本的 dilation 方法，在原陣列圖形中存在實物之點(RGB!=0 處)，作為
為中心點疊合 kernel 延伸找出最大值，並將該中心點值利用擂台法替換為該
最大值存入新陣列中。

2. Erosion



Code fragment:

```
def erosion(img,kernel):
    arr = np.zeros(img.shape, np.uint8)
    for i in range(0, img.shape[0]):
        for j in range(0, img.shape[1]):
            count = True
            min = np.iinfo(np.int64).max
            for ite in kernel:
                if(count==False):break
                x, y = ite
                if x+i < 0 or x+i>=img.shape[0] or y+j<0 or y+j>=img.shape[1]:
                    count=False
                    break
                if img[i+x][j+y]==0:
                    count=False
                elif img[i+x][j+y]<min: min=img[i+x][j+y]
            if count == True: arr[i][j]=min
    return arr
```

以 structing element 之中心為基準，尋找原始圖片陣列中與 structing element 疊合後完全相同之區域，並以擂台法找出該疊合範圍之最小值後，將中心設為該最小值。

3. Opening



Code fragment:

```
def opening(img, kernel):  
    arr = np.zeros(img.shape, np.uint8)  
    arr = erosion(img, kernel)  
    return dilation(arr, kernel)
```

根據課本定義： $f \circ k = (f \ominus k) \oplus k$

先執行 erosion 後執行 dilation 即為 openings

4. Closing



Code fragment:

```
def closing(img, kernel):  
    arr = np.zeros(img.shape, np.uint8)  
    arr = dilation(img, kernel)  
    return erosion(arr, kernel)
```

根據課本定義：

$$f \bullet k = (f \oplus k) \ominus k$$

先執行 dilation 再執行 erosion 即為 closing