

## 電腦視覺 CV Home Work 5 資研一 R07922003 劉濬慶

此次作業跟 hw4 極為類似，只是圖改成灰階圖進行型態轉換以及 Dilation 和 Erosion 有一點改變。

### ● Dilation



掃過 灰階圖 上每個 pixel，若  $\text{pixel} > 0$ ，設此 pixel 的點為  $a$ ，則以  $a$  為原點，C 掃過 kernel 上所有點，將值為 1 的點，對應到 灰階圖 上的 pixel 點 save 在  $\text{pixel\_list}$  中，再將  $\text{Max} = \max(\text{pixel\_list})$ ，之後輸圖上的對應的 pixel 點設為  $\text{Max}$  即可

### ● Erosion



掃過 灰階圖 上每個 pixel，設掃的那點 pixel 的為  $a$ ，則以  $a$  為原點，與 kernel 的原點疊，掃過 kernel 上所有點，將值為 1 的點，對應到 灰階圖 上的 pixel

的 save 在 pixel\_list 中，再將  $\text{Min} = \min(\text{pixel\_list})$ ，在 灰階圖 的位置中“全部都  $> 0$ ”，才將輸出圖上的 pixel 設為 Min 即可

#### ● Opening



先 Erosion，再 Dilation

#### ● Closing



先 Dilation，再 Erosion

#### ● 程式碼:

```
def dilation(binary,kernal,coulmn,row):  
    pix=binary.load()  
    img_new=Image.new(binary.mode, binary.size)
```

```

for i in range(coulmn):
    for j in range(row):
        if pix[i,j] > 0:
            pixel_list=[]
            for x in range(-2,3,1):
                for y in range(-2,3,1):
                    if kernal[x+2][y+2] == 1:
                        if (i+x >=0) and (j+y >= 0) and (i+x < coulmn) and
(j+y < row):
                            pixel_list.append(pix[i+x,j+y])

            Max = max(pixel_list)
            for x in range(-2,3,1):
                for y in range(-2,3,1):
                    if kernal[x+2][y+2] == 1:
                        if (i+x >=0) and (j+y >= 0) and (i+x < coulmn) and
(j+y < row):
                            img_new.putpixel((i,j),Max)

            #img_new.save('dilation.bmp')
            return img_new

def erosion(binary,kernal,coulmn,row):
    pix=binary.load()
    img_new=Image.new(binary.mode, binary.size)
    for i in range(coulmn):
        for j in range(row):
            draw=1
            pixel_list=[]
            for x in range(-2,3,1):
                for y in range(-2,3,1):
                    if kernal[x+2][y+2] == 1:
                        if (i+x < coulmn) and (j+y < row) and (i+x >= 0) and (j+y
>= 0):
                            pixel_list.append(pix[i+x,j+y])
                            if pix[i+x,j+y]==0:
                                draw=0
            else:
                draw=0

```

```
        Min = min(pixel_list)
        if draw == 1:
            img_new.putpixel((i,j),Min)
    #img_new.save('erosion.bmp')
    return img_new
```

```
def opening(binary,kernal,coulmn,row):
    p=erosion(binary,kernal,coulmn,row)
    op=dilation(p,kernal,coulmn,row)
    #op.save('opening.bmp')
    return op
```

```
def closing(binary,kernal,coulmn,row):
    p=dilation(binary,kernal,coulmn,row)
    close=erosion(p,kernal,coulmn,row)
    #close.save('closing.bmp')
    return close
```