

Gray-scaled Morphology

在做出下列四個效果之前，一樣要先將 `lena.bmp` 的原始圖檔用 `Image.open` 讀進來，再來將 octagonal 3-5-5-5-3 kernel 用 array 的方式簡單地刻出來。

(a) Dilation



這邊我們用一開始所得 `lena` 的長寬進行迴圈來做處理，一開始先設一個暫時為 0 的 `temp_Max`，接下來由 `kernel` 的 `element` 的值一個一個進行擴張，只要沒有超出圖的範圍，則去取原圖擴張後新的點的 `pixel` 值，再來和 `temp_Max` 比對之後，留下大的值成為新的 `temp_Max`，做完一輪之後，將 `Max` 值放入和原圖相對應的位置上，就可以進行下一個迴圈的比對。最後結果如上圖。

(b) Erosion



再來 erosion 的部分，和 dilation 大同小異。不同的地方為，相較於 dilation 為保留較大的 pixel 值，erosion 是在每一輪保留較小的 pixel 值，最後同樣地將 min 值放入和原圖相對應的位置，即可得出上圖

比對環節的 code : (上下分別為 dilation 取 Max 和 erosion 取 min):

```
temp_Max = 0
for element in kernel:
    x, y = element
    if (0 <= i+x <= image_original.size[0]-1) and (0 <= j+y <= image_original.
        size[0]-1):
        pixelvalue = image_original.getpixel((i+x, j+y))
        temp_Max = max(temp_Max, pixelvalue)
image_dilation.putpixel((i,j),temp_Max)

temp_min = 255
for element in kernel:
    x, y = element
    if (0 <= i+x <= image_original.size[0]-1) and (0 <= j+y <= image_original.
        size[0]-1):
        pixelvalue = image_original.getpixel((i+x, j+y))
        temp_min = min(temp_min, pixelvalue)
image_erosion.putpixel((i,j),temp_min)
```

(c) Opening



再來 opening 的部分則是遵照公式，由 original image 對 35553 kernel

先做 erosion 的動作，再做 dilation 的部分，即可得出上圖效果。

(d) Closing



而 closing 的部分也是遵照公式，original image 對 35553 kernel 先做

dilation 的動作，再做 erosion 的部分，即可得出上圖效果。