

HW#4

R09631050 生機碩一 詹閔棋

(a) Dilation



要對二值化的 lena 做膨脹，我先建立一個與原圖同 shape 的矩陣，接著在雙迴圈內對每一個 pixel，偵測若 pixel 值大於 0 才操作，代表只對白點操作，在 kernel 範圍內，若 kernel 沒有超出矩陣 size，該點就變為白點，將結果儲存到新建立的矩陣內，最終得出結果。

(b) Erosion



此題做侵蝕，一樣先建立一個與原圖同 shape 的矩陣，在雙迴圈內對每一個

pixel，先設定每一個值都為 True，接著在 kernel 範圍內，偵測只要上下左右任何一邊超出邊界，或是 kernel 範圍有碰到黑點，就將該點值改為 False，最終只有值為 True 的為白點，將資料儲存到新矩陣，就可得出結果，這邊沒辦法像做 Dilation 時那樣直接判斷條件後給值，而改成先儲存成 True 與 False 型態的原因是因為當 kernel 範圍超出原圖 shape 時，要直接儲存 value 到新矩陣就會超出新矩陣的 size，導致 error。

(c) Opening



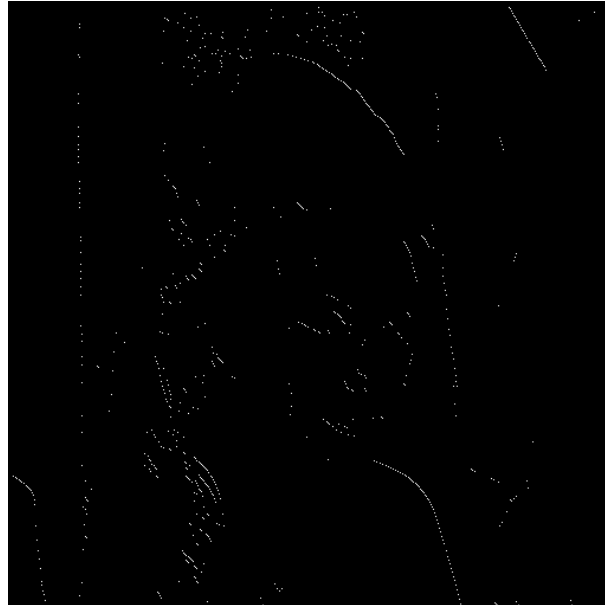
開啟是先侵蝕後膨脹，這邊直接對二值化 lena 套用上面寫好的侵蝕跟膨脹 def，可得到結果。

(d) Closing



關閉跟開啟相反，是**先膨脹後侵蝕**，一樣直接對二值化 **lena** 套用上面寫過的膨脹和侵蝕 **def**，可得到結果。

(e) Hit-and-miss transform



此提按照上課教的公式下去處理，先**建立 J-kernel 與 K-kernel**，這兩 kernel 的聯集部分是**空集合**，接著將 J-kernel 對二值化 **lena** 做侵蝕，將結果儲存成圖 1；再把二值化 **lena** 做出它的**補集**，**黑白點全相反**，將結果存成圖 2，接著用 K-kernel 去對圖 2 做侵蝕，結果儲存成圖 3；建立一個新矩陣，最後一步就是將圖 1 與圖 3 拿出來處理，取**兩圖白點聯集**的部分存到新矩陣，其餘都是黑點，可得出結果。