

Digital Image Processing Homework 2

R13944043 林詩程

Problem 1:

(a) Motivation and Approach:

仔細觀察 sample1.png 發現，圖上噪點都相當細小，應可用 Open Operator 消除，使用和投影片類似的方法，先對 sample1.png 做 Erosion，再對結果做 Dilation，Mask 選用：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Discussion of Results:

結果如同一開始的預期，基本上所有細小的噪點都消失了，但是 Open Operator 也讓不是噪點的部分變得和原圖有一些不一樣的地方，不過都是相當微小的部分，大致上和原圖還是相當一致。



sample1.png



result1.png

(b) Motivation and Approach:

一開始想使用 Close Operator，即先 Dilation 再 Erosion，嘗試後發現這個方法難以將「DIP 2025」消除，故採用另一種做法。

首先利用類似 Flood Fill 演算法的方法把所有和背景相連通的部分塗白，此時除了物件內部的洞口之

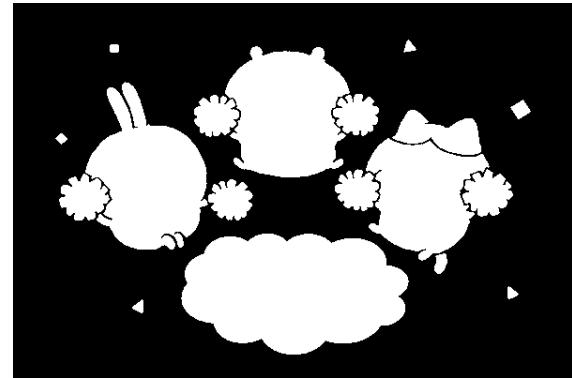
外皆被染為白色，將這個結果取反，此時物件內部洞口皆為白色，而其他部分則為黑色。最後將這個結果和原圖取聯集，即可完美地補上物件內的洞口。

Discussion of Results:

原先使用 Close Operator 時，會輕易地將小八瀏海和尾巴也連通起來，但是難以填補「DIP 2025」，而第二種方法則可以輕易地將「DIP 2025」填補，並分開小八的瀏海和尾巴。然細究所謂「Hole」在一般情況下的界定，應是第二種方法更為妥適，所以我選擇的是第二種方法。



result1.png



result2.png

(c) Motivation and Approach:

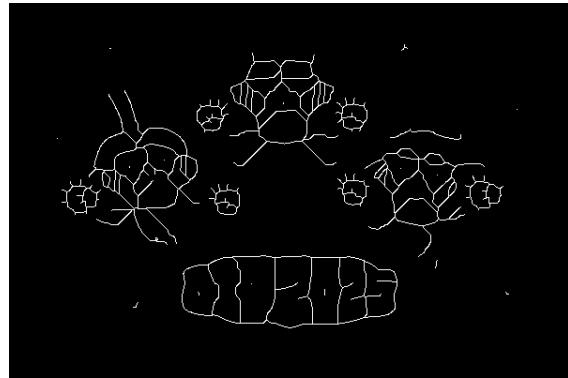
原先不知道該如何處理，後來在網路上找到並採用一個叫 Zhang-Suen Thinning Algorithm 的方法。對所有白色區域（前景）進行兩輪操作，兩輪都先取出前景中一個像素的八個鄰居，並計算白色鄰居的個數和從黑色（背景）轉變為白色（前景）的次數。第一輪中，如果白色鄰居介於 2 ~ 6 之間（排除孤立點和保持主體結構），且轉變次數正好為 1（確保只連通單一區域），加上某些對角與直線方向的像素不能同時為白（避免刪除多個區域連接的橋樑），那就將此點標記為待刪除，最後這輪待刪除的點變為黑色，進行第二輪處理。第二輪大致上步驟和第一輪一樣，惟「某些對角與直線方向的像素不能同時為白」的條件有些許不同。

Discussion of Results:

效果相當不錯，感覺每個物件都被最大程度的細化，並且圖像主體結構基本上都被完整的連接並保留下來，呈現出有點像化石的感覺。



result1.png



result3.png

(d) **Motivation and Approach:**

原先直接用 result2.png 計算物體數量，不過後來公告說小八的身體和瀏海和尾巴要視為同一個物體，於是原先的算法會多出兩個物件，所以是我先對 result2.png 做前處理，將瀏海和尾巴的部分做 Dilation，讓三者相連，再進行計算。

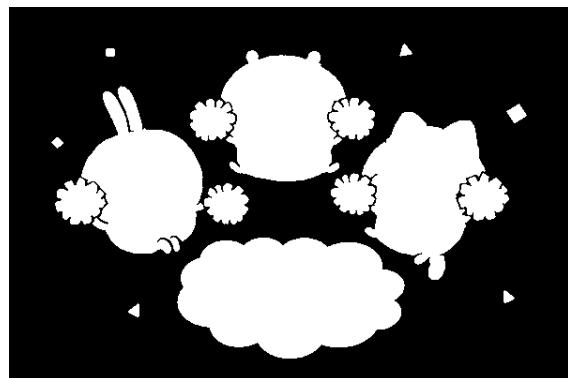
首先對 result2.png 運用類似 Flood Fill 演算法的原理計算總共有幾個連通物體，並將每個物體貼上標籤，找到小八瀏海和尾巴的位置進行 Dilation，結果為 res1d.png，再對 res1d.png 進行一次貼上標籤的步驟，最後得出的標籤數扣掉 1（背景）就是最後的結果。

Discussion of Results:

經過計算，最後得出的物體數量為 16，包含三個生物，六顆彩球，一朵雲，以及六個幾何圖案。結果和肉眼辨別的數量相同，表示計算上應是正確的。



sample1.png



res1d.png

Problem 2:

(a) Motivation and Approach:

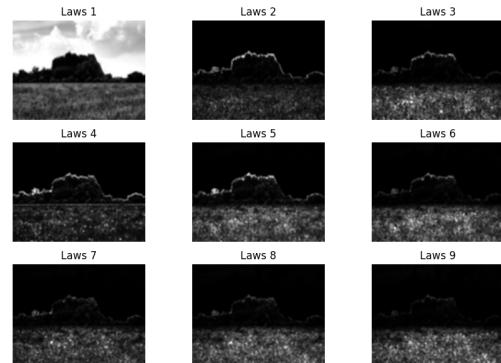
用和投影片上差不多的方法，首先建立投影片上的九個 Mask (Laws 1 ~ Laws 9)，分別對 sample2.png 做卷積，得到 $M1 \sim M9$ 。隨後進行 Energy Computation，建立一個 13×13 且裡面元素全都是 1 的 Window，分別和 $M1 \sim M9$ 的平方做卷積，得到並 Normalize Local Features ($T1 \sim T9$)，最後將結果畫成 Feature Map。

Discussion of Results:

結果如 res2a.png 所示，通過不同 Laws 所呈現的結果都有好好被表現出來，從很多方面來說 Laws 1 感覺最接近原圖，剩下八個都顯得有點暗，特別是天空的部分。



sample2.png



res2a.png

(b) Motivation and Approach:

粗略地觀察 sample2.png，發現大致上可以將圖片分為三類，仔細一點看覺得天空分為兩類似乎更為合適，雲朵和天空放在一類應該不是一個好主意。

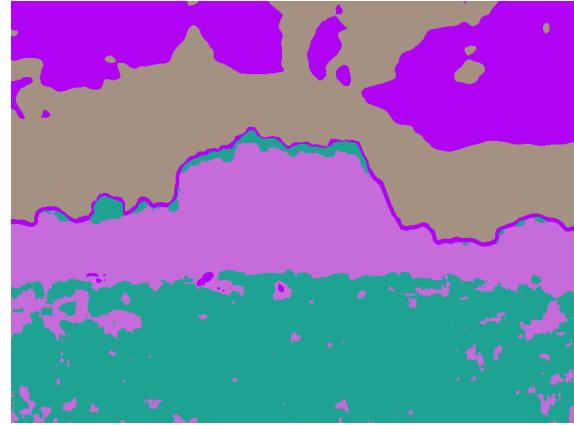
利用上題獲得的 Local Features，總共進行十次迭代，首先選擇隨機不重複的四個點作為中心，計算每個點和四個中心灰階強度的差異，並將其分配到和自己最接近的中心點，之後找出每個群組的平均值作為新的中心點，進行下一次迭代。將最後的四個群組隨機分配四個顏色。

Discussion of Results:

結果如 result4.png 所示，sample2.png 確實有好好被分為四個大類，也都標記上了不同的顏色，和肉眼的辨別差不多，共有較高的樹林、較矮的草叢、天空和雲朵四類，符合預期的樣子。



sample2.png



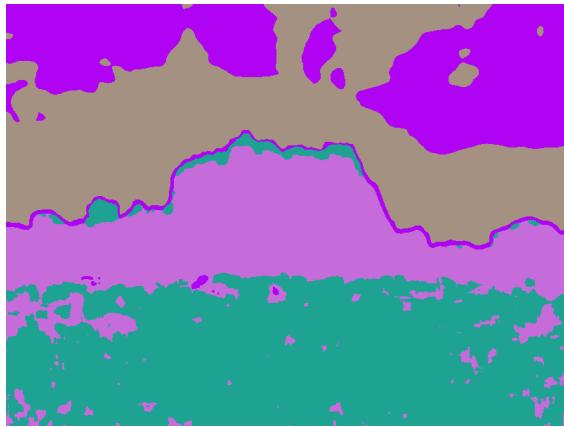
result4.png

(c) Motivation and Approach:

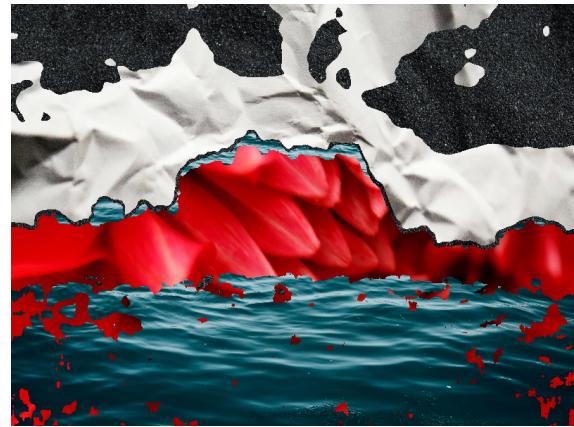
將上題得到的結果改以填充不同的質地，而非不同顏色。我選用紅花填充較高的樹林，選用海洋填充較矮的樹林，並用帶有皺摺的白紙填充雲朵，最後用類似沙子的材質填充天空。

Discussion of Results:

結果如 result5.png 所示，將原本看起來沒有水源的普通大草原，變成富含水資源且充滿想像力的奇幻世界。



result4.png



result5.png

(d) Motivation and Approach:

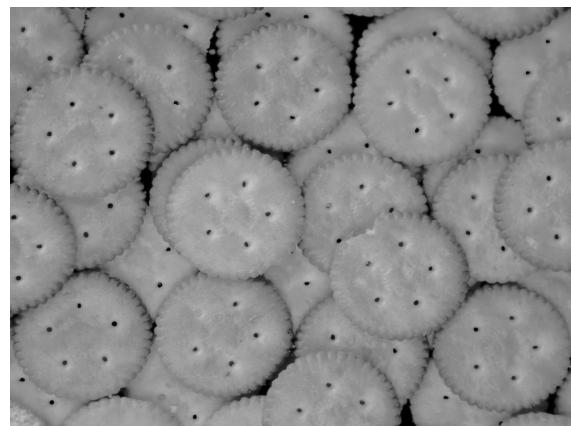
觀察 sample3.png 發現，每個餅乾的大小基本上一致，而我的目標是增加餅乾的大小。

首先經過測試，每片餅乾的大小接近 300×300 ，於是我在 sample3.png 中的一個 300×300 的區塊 400 次，將每次選取的區塊貼上 20×20 方格中的其中一個。

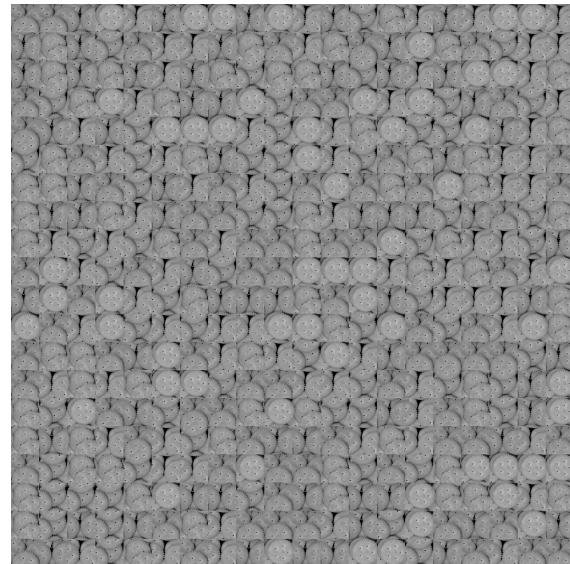
Discussion of Results:

結果如 result6.png 所示，和預期的樣子相去不遠，餅乾變得很多，並且每個區塊內大致上都只有一個餅乾，所以 300×300 算是合適的尺寸。

此外，如果餅乾的數量過少，會造成 Quilting 的痕跡過於明顯，而餅乾的數量過多，則會造成圖片檔案過大，因此兩者間的權衡是一個需要考慮的部分。



sample3.png



result6.png