# Term Project Report

組員:施辰翰、劉樂永

## 一、主題

手寫文字照片的優化處理:將相機拍下的手寫文字圖檔,進行適當的影像 處理,使內容文字更好辨識。成果可能也能作為 OCR 的前製作業。

在這個作品中,我們預設處理的影像是白紙上的黑色或彩色書寫文字。

## 二、動機

生活中有許多機會需要將手寫文字拍成照片,但事後閱讀經常會感到不便。 另外,OCR 技術非常熱門,如何做適當的預處理也值得研究。

## 三、預期結果

正確完成的圖片應該是一個背景全白的影像。上面有清晰的文字,並且將直線取代為電腦繪製的直線。

## 四、預計執行步驟

- 1. 去除雜訊
- 2. 去除晃動模糊
- 3. 分離背景前景
- 4. 將影像銳利化,或其他可能有助於閱讀的優化。
- 5. 辨識出手繪直線,將其移除,換成電腦繪製的直線。

## 五、實際執行步驟

- 1. 去除背景
- 2. 去除淺色雜訊
- 3. 去除小面積雜訊
- 4. 影像轉正
- 5. 去除橫線
- 6. 補回橫線

## 六、執行步驟說明

在程式執行中,需要輸入三個參數,分別是: (最小)字母寬度(也就是 欲保留的最小文字)、power-law transformation 的指數。

#### 1. 去除背景

由於文字在畫面中只佔小部份的較小數值(相較於白紙的 RPG 值都極高),可以使用 median filter 就將文字大致消除,只留下背景。再將背景減去原圖後 反白就能得到結果(1-背景+原圖)。

此處使用和最小字母的寬度同寬的 filter 大小。

#### 2. 加強對比

為了第三步分辨深淺色,在此處需要加強對比。因為目的是在[0,1]的範圍內讓黑色(較小)的數值對比更明顯,所以使用指數大於 1 的 power-law transformation。

因為每張圖片的狀況不同,指數的值必須手動輸入決定。

#### 3. 去除淺色雜訊

用步驟一去除背景後,會剩下很多大面積的淺色雜訊,其中包含少數顏色 較深的雜訊。這一步是檢查圖片的所有點,如果周圍字母寬度的一半距離內沒 有較深的顏色,就填成白色。

#### 4. 去除小面積雜訊

經過步驟二之後,剩下深色的小面積雜訊。判斷所有連通塊,將大小小於 字母寬度平方的連通塊全部移除。

## 5. 影像轉正

將文字反白,使得原先的黑色值較大,白色值較小。接著計算圖片 intensity 的重心位置,以此重心分為左右兩圖,再分別計算左右重心。計算得到的兩側 重心連線的傾斜角度,即是需旋轉的角度。

使用直接旋轉的方式,將圖片邊緣補上白色,大小會增加。

#### 6. 去除橫線

將圖片對水平線做 closing(線長為字母寬度的兩倍),再將-15 度到 15 度的所有結果疊合後就能得到圖片的橫線在哪些位置。將得到的橫線用 opening

(使用字母大小八分之一的圓形)稍微加粗後,再用類似去除背景的方式和圖 片相減就能去除直線了。

#### 7. 補回橫線

由右到左、上到下找出每一條橫線的開頭,DFS 找到距離開頭最遠的點做終點,再標記兩點之間所有被直線切過的格子點,垂直加粗就能畫出直線。最後將直線設為紅色貼上圖片。

加粗的寬度是是字母大小的十分之一。

## 七、執行結果與討論

#### 1. 去除背景

可以看到我們後來使用的步驟順序和原先安排的不一樣,也不符合一般先去除雜訊再操作的步驟。這是因為,在一般的文字圖片中,文字的資訊量實在太少,以至於多數去除雜訊的演算法都會將文字視為要去除的雜訊。舉例來說在(圖)使用邊界偵測的演算法,就只會偵測到背景中那段很明顯的陰影。

因此,我們反過來運用文字的資訊很少的特點,用 median filter 除去文字,就能有效把文字和背景光影分離了。

另外,這些操作在 RGB 空間上分別操作的話都會出現原先不存在的顏色, 所以都要在 HSI 空間上對 intensity 操作。

#### 2. 去除雜訊

在去除背景之後,留下來的雜訊主要是紙張本身帶有的粗糙紋理和去除背景時的誤差(median filter 和原圖的差)。

這樣的雜訊有一種特性,就是很廣泛的分佈在圖片上,淺色居多但也又很多深色顆粒。要去除深色顆粒很簡單,只要是比文字還小的區塊都是不需要的資訊(雖然可能會意外移除標點符號或 i, j 上的點)。但是在淺色區域還大量遍布的時候,就沒辦法這樣操作。

要移除淺色雜訊的話,雖然可以在前一個步驟選擇夠好的 power-law transformation 來增強對比,用 threshold 直接去除淺色雜訊。但是,在書寫文字中,淺色並不都是不重要的資訊。在鉛筆字尤其明顯,在文字旁會有向外擴散的淺色暈染效果和一些筆劃相連的淺色部份,如果直接移除掉會造成視覺上很大的影響。我們的主題是針對人眼辨識而非 OCR,應該設法解決這樣的問題。

後來想到的解答就是,從「文字周圍的訊號比較重要」這點來考量,也就 是當一個淺色訊號附近沒有文字的時候就將其移除。那麼要判斷有文字在附近 也就是透過一開始得到的字母寬度的參數,來偵測這個寬度的一半範圍內是否有顏色很深的資訊。

這樣一來,會留下的資訊只有文字周圍的訊號,還有深色雜訊周圍的區域。 因為當時採用的是字母寬度的一半,所以留下的雜訊大致上都會比一個字母的 面積還小,就能以此為條件剔除了。

#### 3. 影像轉正

這部份是參考自 NEW PREPROCESSING TECHNIQUES FOR HANDWRITTEN WORD RECOGNITION (M. Blumenstein, C. K. Cheng and X. Y. Liu)中的方法。

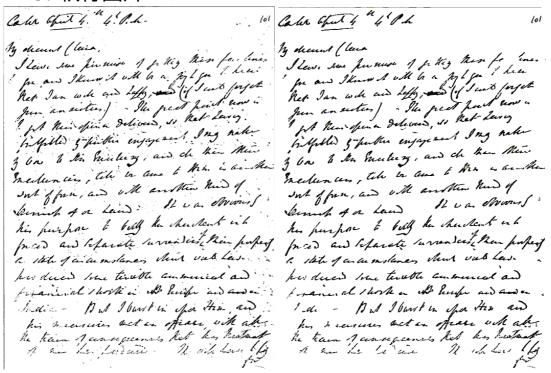
只要求出文字圖片的重心,再將圖片依此重心分成左右兩部份,各找出一個重心。左側重心和右側重心的連線,其傾斜角度就是應該旋轉的角度了。

但是這部份在某些特殊文字會嚴重失效,例如文字分佈有特殊狀況的書信 或書本。在單一一行文字時表現是很好的。

#### 4. 線條偵測與替換

偵測線條是使用很單純的 closing 操作。一般來說這種狀況應該使用 hit and miss,但是這種方法比較難在各式各樣大小的圖片裡自動使用,也難以應付線條粗細差距很大的狀況,就不使用了。

### 5. 執行圖片

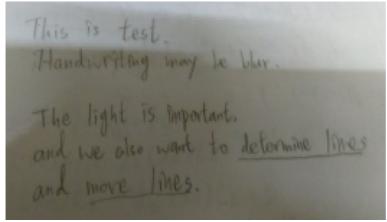


書寫體的污點可以去除,但也會去除掉很多內容。(左為原圖)

Cale Chart & a for the form of the form of

Caler april 4 "4' Ph

反歪斜的結果很差(左)、線條判斷(右)也總是判斷到文字內部



嘗試處理非常模糊的圖片

This is test.
Handwriting may be there.

The light is important, and we also want to determine lines and move lines.

去除背景

This is test.
Handwriting may be ther

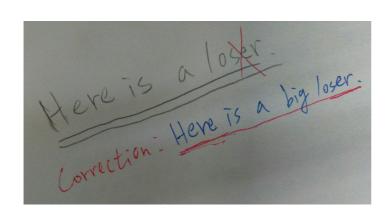
The light is important,
and we also want to determine lines
and more lines.

#### 去除雜訊

This is test.
Handwriting may be ther

The light is important,
and we also want to determine lines
and move lines.

#### 替換線條



Here is a loser Correction: Here is a big loser.