DIP FINAL PROJECT –

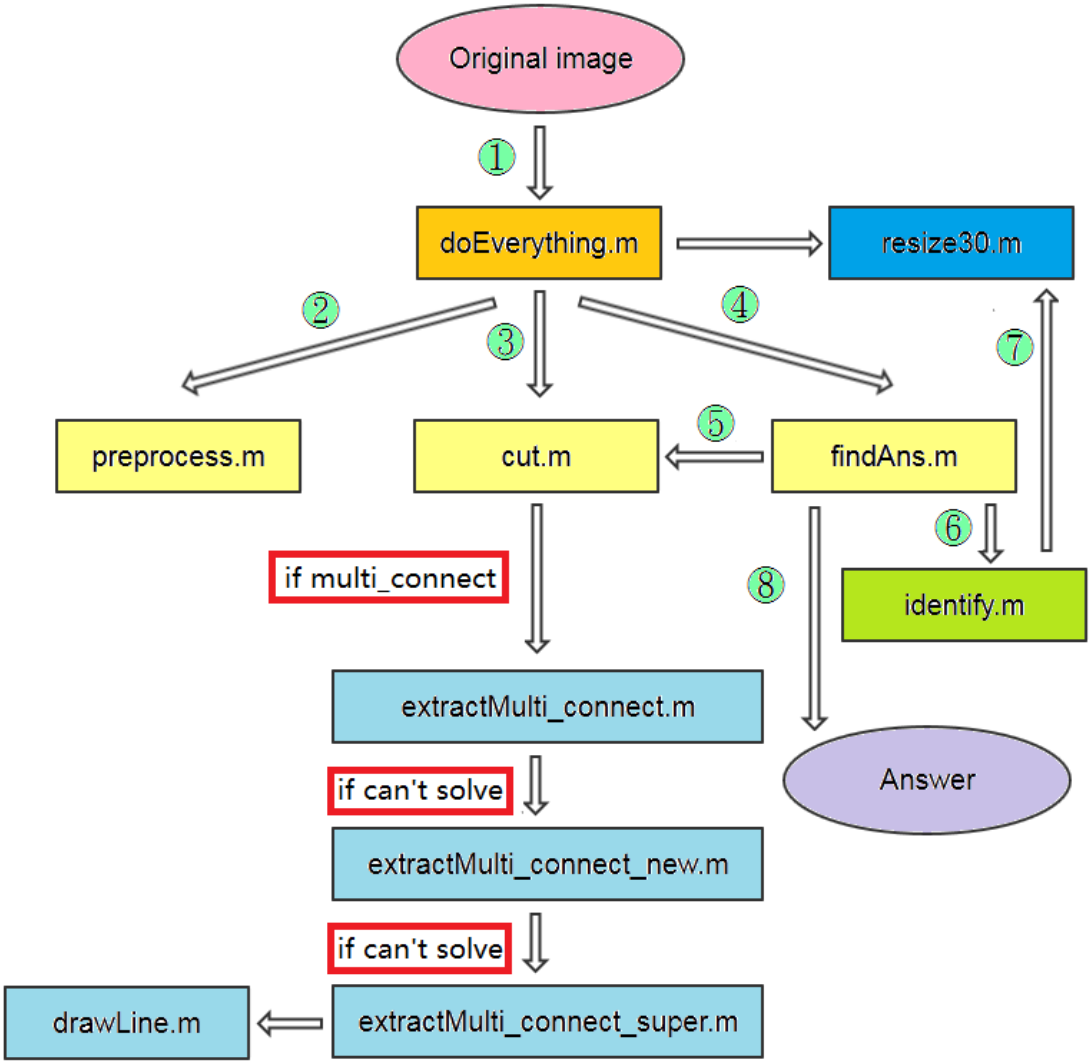
CAPTCHA CRACKING

102062309 王暉博 102062221 史芳瑜

◆ Motivation

俗話說：「科技來自於人性。」身為清大的學生，相信都曾因為點不到喜歡的課而煩惱過。通識排第一還點不到，當準備放棄時，又在加退選時段看到他突然多出了一個名額，卻又因手速比其他人慢而沒搶到。每天守在電腦前刷報表，心情像是在洗三溫暖一樣，忽冷忽熱。某天，我們不願意再讓選課影響我們的心情，決定發揮資工系的長才，寫一個自動點課機器人來幫助我們點課（一直狂刷報表，當有名額時自動點課），但這計畫卻因無法打敗萬惡的驗證碼而暫緩。為了完成我們遠大的夢想，在三上這個爆炸的學期，我們毅然決然的點了影像處理這堂課。經過老師一個學期的諄諄教誨，我們學會了許多影像處理的技巧，讓我們在破解驗證碼的途中事半功倍！以下便是我們實作的方法。

◆ Implementation



1. 將原圖傳入 function doEverything.m 中。在 doEverything.m 中建立字模，並傳入 resize30.m 做 padding，將各字模變成 30x30。
2. 將原圖傳入 preprocess.m 把圖片去雜點和轉成二值化圖片後，回到 doEverything.m。
3. 進入到 function cut.m。cut.m 是這個 project 的精髓，目的就是把在圖片上的字一一切出來。而當遇到數字相黏的情況時，就會進入到 extractMulti_connect.m，若 extractMulti_connect.m 無法成功將字切開，便會進入到 extractMulti_connect_new.m，若再失敗，則進入 extractMulti_connect_super.m。extractMulti_connect_super.m 還會 call drawLine.m 來輔助他。若無數字相連的情況便會單純的把數字都切開再回到 doEverything.m。
4. 把切好的各數字和字模傳入 findAns.m，對數字做旋轉。

5. 把旋轉完的圖片傳入 cut.m，把不需要的部分切除。
6. 把字模和旋轉完的三個數字傳入 identify.m。identify.m 主要是將各數字和字模做比對，取相似度最高的作為答案。
7. 將數字傳入 resize30.m 做 padding，將各數字變成 30x30。
8. 得到答案。

◆ Detailed Method

■ Image Denoising

首先，我們將彩色的圖片 [1]轉成灰階圖[2]以方便實作，再根據觀察得知雜點顏色都較深，因此將圖片 [2]灰階值 20 以下的都設成 0，便可得到乾淨無雜點的圖片 [3]。順便將圖片二值化[4]以方便作業。

Original Image[1]

Gray Scale Image[2]

Image Denoising[3]

Image Binarization[4]

■ Building Disjoint Sets

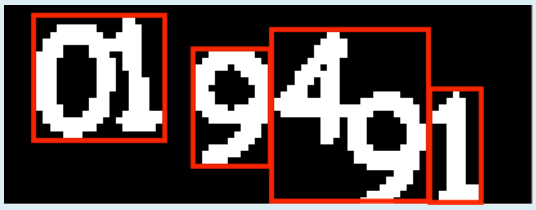
得到二值化的圖片後，我們就要想辦法將字一個一個切開。一開始上網 google 到 matlab 內建函式 bwlabel，他可以在一張 binary 圖片中找出每個 disjoint set，將他們用不同 label 區別。bwlabel 還可以選擇要用 4-connected object 還是 8-connected object，而在這裡我們是選用 4-connected object。當各數字被不同的 label 區別後，我們就記錄他左上和右下兩個點，將他們從原圖中切出來。下圖為示意圖。

Original Image

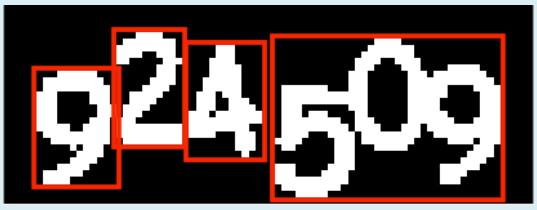
Crop Image

Result

然而事情常常不盡如人意，當我們以為這樣就可以完成時，才發現大部分的圖片都不是這麼的單純。他們大致上都長得像下圖。



2-connected components[5]



3-connected components[6]

4-connected components[7]

5-connected components[8]

有兩個數字連在一起的[5]，到三個[6]、四個[7]、甚至有五個[8]數字黏在一起的狀況。因為這些都無法成功用 bwlabel 將各個數字分開，所以我們要另覓他法。

根據觀察，我們發現到數字的灰階值是有規律的：每個數字的灰階值比他們右邊的數字還要小。因此我們可以根據數字的灰階值來將數字們分割。數字相黏大致上可分成三種情況，每種情況有不同的應對方法。

1. 數字間有雜點導致灰階後數字相連（用 extractMulti_connect.m 可解決）

Original Image

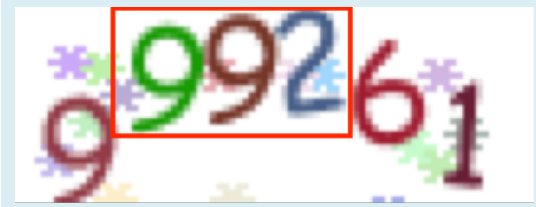
Gray Image

這個情況是最常見，也是最容易處理的。可以看到左圖在彩色時紅色圈起來的部分是不相連的，但在轉為灰階影像後，卻因為雜點沒有去除乾淨的關係而導致數字相黏。（註：因為我們是用 threshlod<20 的方法把雜點去除，但若雜點出現在數字和數字之間，灰階值往往會比 20 還要大，導致雜點無法準確地去除。）

解決方法：

根據我們的觀察可以發現，在灰階圖上 4 跟 6 有明顯的色差，所以我們將 4 與 6 這個相連的 component 獨立出來，觀察 histogram 上 4 跟 6，兩個數字的灰階平均分布分別對應兩個波峰，利用 local Otsu's multi-thresholding 即可輕鬆將兩個數字給分開。

2. 數字間輕輕相觸（用 extractMulti_connect_new.m 可解決）



Original Image

Gray Image

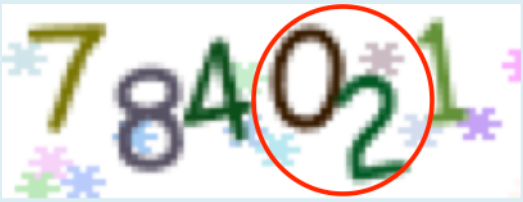
這個情況最少發生，難度中等。主要是因為圖片的解析度太低，導致數字邊框外圍會有模糊的現象，而當轉成灰階圖按時數字就相連在一起了。

解決方法：

在這個情況下我們一樣觀察了 histogram，發現在這種情況下利用上述的解決方法並沒有辦法很有效的將數字給分開，這是因為 histogram 的分布較分散且沒有明顯的波峰，所以我們利用了

gamma transform 將數字 mapping 到較狹窄的 range 裡，藉由這種作法，使用 Otsu's method 後可以得到較好的結果，經過 errosion 與 dilation 後即可得到足以正確分析且獨立的數字圖片 。

3. 一個數字疊到另一個數字上（用 extractMulti_connect_super.m 可解決）



Original Image

Gray Image

這種情況也蠻常發生的，前面兩種我們覺得都不是設計這個驗證碼的人會預料到的事情，而這種情況則應是有意為之。簡單來說就是一個數字覆蓋到另一個數字上，導致兩個數字交界處灰階值特別低。

解決方法：

根據我們的實驗，如果不做任何處理直接套入 Otsu's method 得到的結果往往是只有數字的邊框而無法得到完整的數字，我們推測這是因為兩數字相疊交界處灰階值的異常導致 Otsu's method 得到的 threshold 不好，所以我們使用了 canny edge detector 將數字的邊框找出來，再從原圖減去，這樣可以得到一個去邊的數字 component，藉此降低邊框對於 histogram 的影響。這種暴力做法讓我們在大多數的時候都能得到正確答案，但在一些少數情況仍會有錯誤，我們猜想我們成功率大概就是卡在這個情況無法百分百準確地解決。

■ Image Recognition

首先，我們建立標準字模，也就是我們期望數字所長的樣子，如下圖。

Standard number

接著，由於不是每個數字切出來長得都像字模一樣端正，他有可能會歪斜，因此我們對每個數字做順時針和逆時針各轉 10 度，再拿原圖和這兩張圖片去和字模做比對，取符合度最高的當作這個數字的答案。另外，由於每個數字的字模大小不一，因此我們對字模以及三個要比對的圖片做 padding，統一大小為 30x30。

◆ Results & Analysis

Original Image

Answer

我們這個 project 在做出第一版（沒有 extractMulti_connect_new.m 和 extractMulti_connect_super.m）時就已經有了 8 成左右的成功率。我們有想過再去破解其他系統的驗證碼（像是台鐵），但最後我們認為應該把重心放在提高我們 project 的辨識成功率。於是我們花了很多的時間，加入了 extractMulti_connect_new.m 和 extractMulti_connect_super.m 這兩個 function，終於將成功率提高到 9 成 5 以上。在測試用的 100 多張驗證碼中，只有 5 張不能夠被成功辨識，而不能夠被辨識的情況通常是有數字疊到另一個數字上。老師在我們報告時有提到如果把使用彩色的圖片是否可以解決這個問題？由於這個驗證碼再抓下來後丟進 matlab 時就是灰階圖片了，後來我們有找到將圖片轉回彩色的方法(在 imread 時多吃一個 argument map，再用 ind2rgb 就可以將灰節圖片轉為彩色)，但我們認為這無法有效提高我們的成功率，因為剛剛提到，不能夠被辨識的情況通常是有數字疊到另一個數字上，若是用彩色圖片下去實作，很有可能導致切割後的數字有殘缺，而導致無法辨識。

◆ Future Work

我們的 Final Project 雖然只是一個小小作品，但他可以有許多有趣的應用。由於 matlab 笨重且速度緩慢，之後我們會考慮將他改寫成 java 或是 python 版本，製成簡易的小工具，可以直接紀錄帳密，一鍵登入校務資訊系統，又或者是寫成選課 APP，免除校務資訊系統不支援行動版這個缺陷，讓使用者能更輕易地使用手機登入校務資訊系統查看成績、修課同學，甚至是很輕鬆的就能在行動終端選課。另外我們發現加退選系統的驗證碼組成結構與校務資訊系統登入時的驗證碼結構極為相似，只有字數不太一樣，或許藉由我們 Final Project 可以完成一些很有趣的功能，這正是我們修習這門課的初衷，也是我們最大的收穫，期望能為大家的大學生活帶來更多的便利。

除了 Final project 的應用外，我們也想要嘗試破解有更複雜機制的驗證碼。如下圖，該圖中有許多的噪音線、扭曲的字體等，這都是我們有興趣繼續嘗試的題材。