****

數位影像處理 期末書面報告

指導教授：張麗娜 副教授

題目：基於 MATLAB 驗證碼識別

學生:

謝維澤 0086C205

許智程 01053303

中華民國一一二年一月二日

目錄

一、簡介………………………………………………….3

二、工作流程圖…………………………………….4

三、方法 ……………………………………………….5

四、最終執行果…………………………………..26

五、心得感想……………………………………….27

六、參考文獻………………………………………28

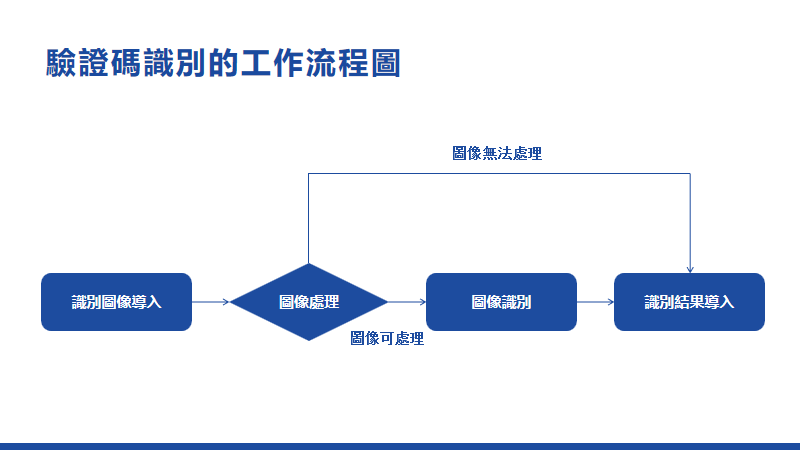
1. **簡介**

為什麼要做這份專題呢?原因是因為我們常常覺得網路上的驗證碼有時候實在是太花俏了，導致常常都輸入錯，連我們這種年輕人都會有看不清楚的時候，那就更別說那種有深度近視，甚至是有老花眼的中老年人了，對他們來說可能偶爾想登入個帳密加輸入驗證碼去網購，結果卡在重新輸入這邊進不去實在太痛苦，所以我希望我能利用這學期上課所學的方式，透過課堂中的圖片分割、轉黑白、膨脹腐蝕、開閉運算…等方式，來達到預測的效果。

減少要一再重新輸入驗證碼的冏境，不但能將上課的內容做到學以致用，更遠大的目標是，能造福社會，為社會盡一份微薄的心力。

1. **工作流程圖**

本次專題以MATLAB實現，實現讀取圖片、讀取灰度值等圖像處理相關功能。圖像處理及識別的主程序的總體流程如下：導入（讀取）驗證碼圖片，圖像處理，圖像識別，識別結果輸出。



1. **方法**

**法一** **:** **replace those color equals to**

**rectangle edge color to white**

**想法 :**

1. 利用 find 函數提取目標，獲取其索引
2. 再根據索引，進行著色渲染處理

**執行方式**

• **先令上下水平上的邊和左右垂直上的邊，並將它們各自連結再一起，利用 concat 方式**

H = cat(2, I\_rgb(1,1:width), I\_rgb(height,1:width));

V = cat(1, I\_rgb(1:height,1), I\_rgb(1:height,width));

• **再將 width 的長度連結再一起，變成 2 個width 和 2 個 height**，

• **其中.'為轉置，轉置之後相加會變成 1\*2w+2h**

• **用來分割長和寬**

side\_colors = cat(2, H, V.');

• **將重複的刪除**

side\_colors = unique(side\_colors);

I\_clean = I\_rgb;

• **每次都使用一個 side\_colors (四個邊上面的顏色) 來偵測，如果有一樣的就變成白色**

**for 的作用就是把邊上特定顏色跑一遍**

for i=1:length(side\_colors)

• **find 作用為找 index (i\_clean裡面的特定顏色) 255:全白**

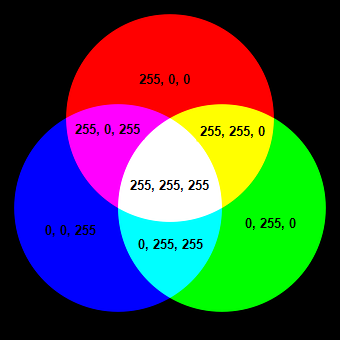
I\_clean(find(I\_clean==side\_colors(i)))=255;

end

if Nth\_DATASET > 4

• **覆蓋**

I\_clean = I\_rgb;

 end

**執行結果**

**Before**



**After**



**法二** **:** **automatically select a threshold to**

**denoise image**

**簡單介紹 :**

原先彩色圖像共有 256\*256\*256 種顏色變化，而灰

度圖像只有 256 種，將其轉換的目的能讓圖片依然

清楚，同時能減少後續的計算量，達到易讀效果。

**想法 :**

(1) 先將 RGB 圖像或顏色圖轉換為灰度圖

(2) 找灰階圖的 threshold

(3) 用 threshold 把灰階圖轉黑白

**執行方式**

• 先將 RGB 圖像或顏色圖轉換為灰度圖

I\_gray = rgb2gray(I\_clean);

• 閾值分割,使用 Otsu 方法计算全圖像閾值，最大類

間方差法

threshold = graythresh(I\_gray);

if Nth\_DATASET == 0

threshold = 0.4;

End

• 使用 imbinarize function 將所有高於全局閾值

的值替換為 1 並將所有其他值設置為 0 ，從二維

或三維灰度圖像 I 創建黑白圖像。

I\_thresh = imbinarize(I\_gray, threshold);

**執行結果**

**Before**



**After**



**法三 : construct a square structure**

**element to make image dialate or**

**make image open**

**簡單介紹:**

膨脹為將與物體接觸的所有背景點合併到該物體中，

使邊界向外部擴張的過程。可以用來填補物體中的空

洞。

開運算為先侵蝕再膨脹，消除小物體在分離，跟侵蝕

很像，差別再膨脹、侵蝕都會改變面積，開運算不會

開運算目的是把因為膨脹而不小心黏再一起的地方分開來

**想法:**

(1) 反色

(2) 再將圖片放大

(3) 對欲膨脹的區域設置一個方形，然後進行膨脹

(4) 縮小

(5) 做開運算

**執行方式**

**把變成黑白的圖片做反色(黑白變成白黑)**

**% 輸入 0 返回 1 輸入 1 返回 0**

I\_reverse = (I\_thresh ~= 1);

**%~= == !=% 先放大 1.25 倍% 變成3\*3 的方形**

I\_dilate = imresize(I\_reverse, 1.25); SE0 = strel('square', 3); **% 膨脹**

I\_dilate = imdilate(I\_dilate, SE0);

**% 在縮小 0.8 倍，如果沒有先放大在縮小的話，圖**

**片會變得很誇張 % 膨脹**

I\_dilate = imresize(I\_dilate, 0.8); I\_dilate1 = imdilate(I\_reverse, SE0);

**% I\_open = imopen(I\_reverse, SE1);**

SE1 = strel('disk', 3);

**ones(5,1)為垂直，ones會創造裡面都是一的矩陣，**

**去比較，如果線蓋到的地方會計算裡面一的個數，如**

**果小於某個值(垂直連接不夠的地方)就會設為0**

I\_open = imopen(I\_reverse, ones(5,1));

I\_close = imclose(I\_open, ones(1,5)); **%%先不用**

subplot(4,2,1), imshow(I\_rgb), title('RGB image')

subplot(4,2,2), imshow(I\_clean), title('clean image')

subplot(4,2,3), imshow(I\_thresh), title('threshold image')

subplot(4,2,4), imshow(I\_reverse), title('reverse image')

subplot(4,2,5), imshow(I\_dilate), title('dilate image')

subplot(4,2,6), imshow(I\_open), title('open image')

subplot(4,2,7), imshow(I\_close), title('close image')

subplot(4,2,8), imshow(I\_dilate1), title('dilate1')

**執行結果 - 膨脹**







**執行結果 - 開運算**







**執行結果 (如果沒有先縮小再膨脹的結果)**

**縮小後膨脹再放大**



**直接膨脹**



**法四 : cutting image into several**

**characters**

**簡單介紹 :**

將驗證碼圖片上的字進行分割和存取

**想法 :**

(1) 將法三中反色後的圖拿來用

(2) 將圖中上下左右不重要的地方給去除掉

(3) 存切下來的字卡跟切完的驗證碼圖

(4) 設定驗證碼自數，大於等於 4 才會停止

(5) 如果切了一定次數都沒有成功，則做開運算

**執行方式**

while true

i = 1;

I\_crop = crop(I, cnt);

**%把上下左右不重要的地放給去除掉**

while size(I\_crop,2) > 10

**%%如果寬太長的話就會依值執行下面的迴圈**

[character, I\_crop] = get\_next\_char(I\_crop, cnt);

**%抓下一個character，[]裡面一個是已切除的圖，另一個是尚未切除的圖**

if size(character,1) > 0 && size(character,2) > 0

**%如果 character 的寬跟高都 >0 的話 %看是否為空矩陣**

chars{i} = character;

**%如果不適空矩陣的話，令一個數組來儲存他**

**i = i+1;%在往下一個圖片去抓**

else

continue

disp(length(chars)); **%看切出來有幾個character**

if length(chars) >= 4

**%因為一般來說驗證碼都>=4個數**

Break

end

cnt = cnt+1;

**%如果沒有的話，令一個計數器去+1，計數器每增**

**一個就是增加一次容忍值**

if cnt==20 || open

**%cnt就是我丟進去的n\_divid(想成容忍度)**

open = true;

**%如果切了20次都沒有辦法>4的話，就做開運算**

cnt = 0;

I = imopen(I, ones(3,1));

**%(3,1)每次切一點點點**

**%I = imclose(I, ones(2,1));**

for i=1:length(chars)

chars{i} = imresize(chars{i},[40,30]);

**執行結果**







**超出設定的切除次數**







**法五 : compare cutting characters with   
 standard characters**

**簡單介紹 :**

將我切下來的圖，跟我資料夾裡的圖進行比對，並判斷出最有可能是啥數字或英文後輸出。

**想法 :**

1. 因為 matlab 裡面檔名開頭為 1 的話會優先讀

取，所以我在取擋名時會利用 ascii 碼來命名

(2) 把驗證碼中的字先讀出來，並將路徑拼起來

(3) 將切下來的字卡進行比較並計算 sum 值，最後

挑選 sum 值最大的作為最可能為的結果

(4) 將挑選出來的值的擋名 (ascii) 轉成數字後，利用

char function 轉成 character，並且 output

出來

**執行方式**

**% method 5 : compare cutting characters with standard characters**

k = 1;

if Nth\_DATASET < 4

n\_characters = 33;

**%matlab裡面開頭1的話會先讀取，所以 d-z 有**

**23 個+ 0-9 有 10 個 = 33 個**

k = 24;

End

for i=1:length(chars)

**%把驗證碼中的每個字卡依序讀出來**

for j = k:n\_characters

**%只會讀24->33(數字就會是0~9)，10 個**

**%跑原本做好的字卡**

filename = strcat('./characters/', fileList(j).name);

**%把字卡的路徑拼起來**

character = imread(filename); **%讀取字卡**

intersection(1,j) = sum(sum(chars{i} == character));

**%原本字卡的 image 跟切下來的字卡的 image做**

**交集(兩個都等於1的話 +1)，兩個圖片如果交集**

**(顏色重疊)的地方越多，sum 越大**

**%如果兩個字卡的pixel都是0(黑)的話，也會加(把**

**兩張圖相似的地方疊起來)**

chars{i}:驗證碼 character:做好的圖卡

 [val, idx] = max(intersection);

**%取最大的數字跟位置**

len = length(fileList(idx).name);

**%讀最大值的數字和位置，讀檔名**

str = fileList(idx).name(1:len-4);

**%把.png或.jpg給去除掉**

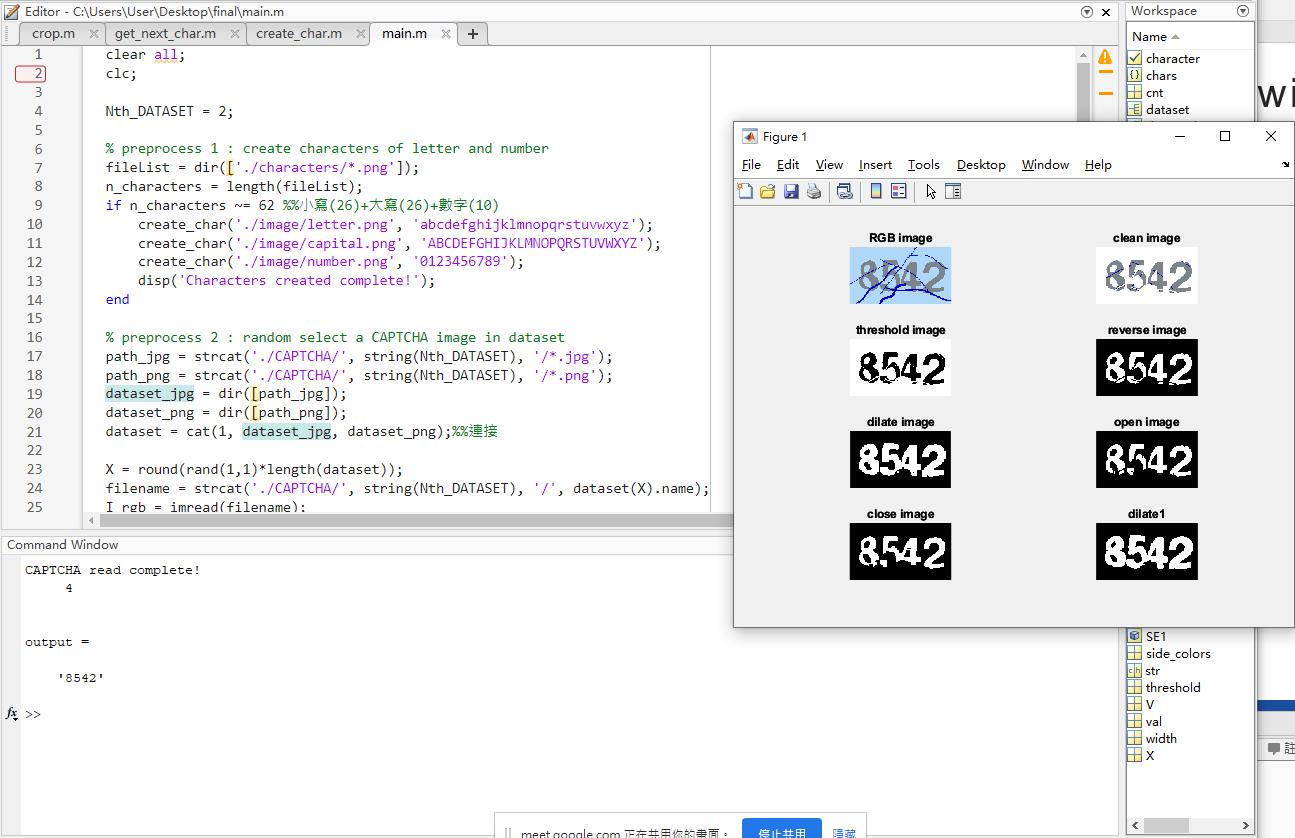
output(1,i) = char(str2double(str));

**%檔名ascii轉成數字之後，再用char function轉**

**成character**

end

四、最終執行結果



1. 心得感想

透過這次的專題製作，我們了解並學習了驗證碼識別的工作原理，這次專題我們運用了許多這學期上課所學的方法來製作，遺憾的是，目前的算法還是很基礎很簡單，因此只能應對這種干擾較少、中規中矩的驗證碼圖片。對於出現了干擾線、驗證碼文字扭曲的圖片暫時未能較好處理和識別。希望在未來能有機會做得更好，不斷改進算法。透過這學期的專題製作，我們受益良多，除了讓我們學習新的知識以外，也能對這學期所學的內容更加了解。

六、參考文獻

•https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.html

•https://ww2.mathworks.cn/help/images/ref/im

binarize.html

•https://www.mathworks.com/help/images/dila

te-an-image.html

•https://www.cs.auckland.ac.nz/courses/compsc

i773s1c/lectures/ImageProcessing-html/topic4.html