

智慧型字元自動辨識演算法

陳鴻桔

國立高雄海洋科技大學

電訊工程系

heavengate@livemail.tw

張玉雯

國立高雄海洋科技大學

電訊工程系

may@mail.nkmu.edu.tw

摘要

本文主要提出一個可以自動辨識發票號碼之演算法，包含發票民國年月號碼辨識及中獎通知。由於發票上年月份和號碼的特徵值不同，因此本研究中首先把發票讀取或影像掃描後利用內定的 2 組閾值分別取出下列 2 種資料：1.發票號碼部分，2.年月份部份。在發票號碼辨識部份 我們利用"垂直/水平投影法"找出其每個字元位置並分割後，再與內定的樣版作模版比對，透過不同加權值的計算來判斷，確認該發票上的號碼。在民國年月部份的辨識也是同樣用"垂直/水平投影法"找出位置後取出年月份可能在的區塊，接著同樣地作模版比對。經實驗結果，平均辨識成功率為 82%，每張發票的平均辨識時間約 1.5 秒。

關鍵字：發票辨識，模版比對，自動

Abstract

The text mainly mentioned about how to auto-recognize counting invoice of number. It included invoice of schedule and how to receive money. Firstly, my research browsed two groups threshold Within some information to pick out two kinds of information: 1. Number of invoice 2. Scheduling (year/ month)

We should recognize number of invoices through differentWeighted for measuring, we use chose Template Matching Method we also employ method of vertical and horizontal standard to find

out. Every position and divided it into two parts.

On the same way, we count account the same Template Matching Method to present schedule (year/ month). Furthermore, after the result of experiment, we almost identified a lot of right cases of 82% and every invoice takes 1.5 second to identify for its average time.

Keywords: identify invoice , Template Matching Method , auto

1. 簡介

統一發票制度是國家的主要稅收來源，提升索取統一發票的誘因是每兩個月一次的對獎活動，獎金雖然不多但也是社福團體主要的資金來源之一，但是由於每月所收集到的捐款發票數量龐大，如果僅由少數人來做對獎的動作不僅費力又費時，人類是透過眼睛去感受顏色進而辨識圖案，但是人眼會疲勞，長時間維持同樣的視角焦距更會造成身體上的不適，所以我們透過電腦來辨識[1]並存入資料庫內，當開獎時只需輸入中獎號碼即可自動對獎。若使用電腦來辨識並自動化對獎[2]不但省時也省力，因此。本文先將每期開獎號碼存入電腦中的資料庫；當對獎時，只需藉由影像擷取設備來擷取發票影像後，利用本文提出之智慧型自動辨識演算法，就可以輕鬆得知是否中獎。避免原先耗時的對獎動作，更能將時間節省下來來更多有意義的事。

2 研究原理

從收集到的發票我們可以先歸納出幾點特徵 1.年月份及號碼都位於發票上方並且在相對位置上不易變動 2.號碼部份各個月分差異不大，字體也統一 3.年月分上方區塊為各廠商自行運用之區塊可能干擾辨識結果。

因此本辦法所使用的基本原理如下：

2.1 利用閾值分割

閾值是利用灰階值 $\geq T$ 則像素轉為白色，灰階值 $\leq T$ 則像素轉為黑色的方式，只要所需要的區域的閾值先測出來之後，利用閾值分割我們就可以將不需要得區域捨棄並把需要的區域獨立出來。[1] [4]

2.2 模版比對

模版比對就像是人眼在辨識物體一樣，首先需對物體觀察過一遍後才會記得他的形狀，顏色等資訊。再次看到此物體時就會和記憶中的樣版作搜尋的動作。當相似度越高時對此物體印象就會越明顯。在辨識圖片上的作法是先將所需的圖像作為模版後 再與被辨識的圖像作比較運算。方法有相減，相關係數等差異值越小表示該辨識圖片與樣版越相似。

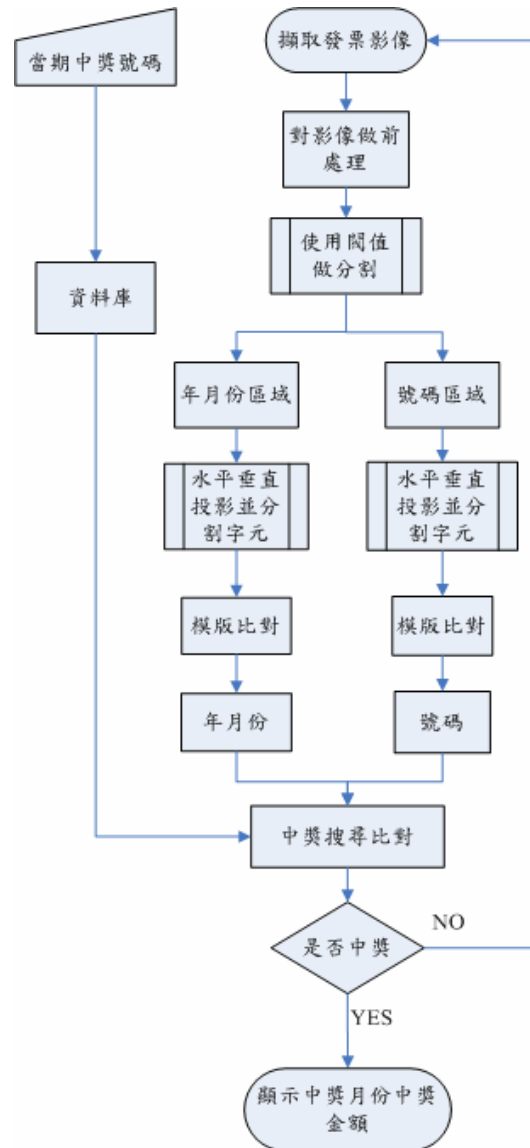


圖 2 本演算法的處理流程

3. 研究方法

一張發票可以分成四部份如圖 1 所示，A 區域是發票最上方至民國年月份的結尾，B 區域是浮水印的上方至下方，C 區域是主要的商品紀錄區，D 區域是各商家的廣告區。在此主要使用 A 及 B 區塊來完成本演算法而整個演算法流程如圖 2 所示。

由統一發票的規格尺寸為寬 4.4 公分，長約 19 公分，因此在發票沒有傾斜的情況之下可以知道 1. 年月份的相對位置大約是在由上往下 5%~10% 的高度，寬度約是由左往右 0.11



圖 1 發票上四個區塊

%~0.79%。另外在A區域中，有時會額外存在店家的商標這對於高度相對位置會有所變更；而影響寬度的變動的因素不多，所以可以把範圍縮小2.發票號碼的所在高度，大約是在整張發票由上往下 0.7%~0.18%的位置，寬度則是由左往右 0.03%~0.75%。因此可利用以上相關資訊先把所要辨識的範圍作縮小的動作，以利速度上的提升。

在縮小偵測範圍之後接著是定位。首先把發票圖片由 RGB 轉變成 HSV 的色彩模型[5]，我們使用V（亮度）來作為分割字元的門檻值，其門檻值設定如下。年月份：[0.85≤亮度值(V)≤0.9] 發票號碼：11-12月 [0.55≤亮度值(V)≤0.75]。

接著，我們利用水平，垂直投影法進一步定位出發票上年月份及號碼的位置座標，如圖3，4所示



圖3 年月份之水平投影圖示

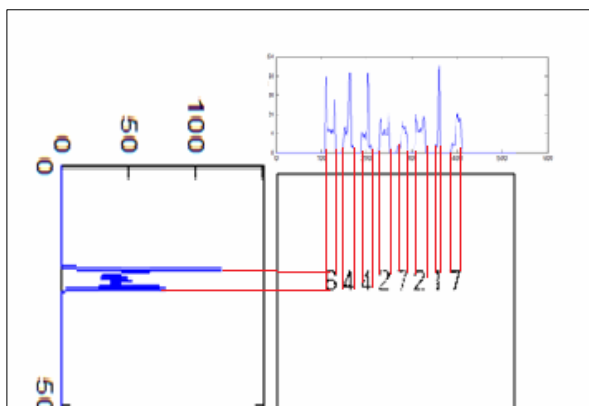


圖4 發票號碼之水平垂直投影圖示

而在年月份定位方面，由於發票左方的黑格會妨礙垂直投影及之後的判斷，這部份以垂直投影可以從發票中間不會受干擾的地方來

進行投影如圖5。

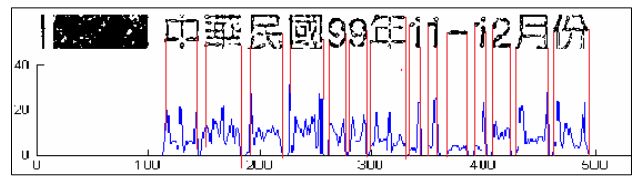


圖5 年月份大區域分割

因此依據投影統計後的資料可得知年月份及號碼的相關位置對應座標並用以進一步來切割出各個獨立的字元，以便進行模版比對。

在模版比對方面分別對各個單張字元模版圖片進行比對。在發票號碼部份因為分割的順序是從左至右，所以在分割出圖片之後將圖片延伸到 48*25 個圖元大小，再分割成 40 等分區塊接著統計各區塊內的圖元值為零的圖元個數為該區塊的值如圖6所示。再進行與模版如圖7相減後乘上相對應區塊的加權值如表1所示。加權值參數的設定是將樣版中沒用到的區塊設定一個大的加權值以拉高數字間的比較差異，而隨著收集的樣本數量的提昇，在可能有些微變動的地方設置較小的加權值如表1。其比對公式如下[2]：

$$\left\{ \sum_{i=1}^{40} [\alpha_{ij} - \beta_{ij}] \rho_{ij} \right\} / 40 = \omega_{wi}$$

$\text{MIN}(\omega_{wi})$ 即是該辨識數字的結果。

α_{wj} ：發票號碼各區塊的特徵值

β_{ij} ：樣版號碼各區塊的特徵值

ρ_{ij} ：模版各區域的加權值

ω_{wi} ：待辨識數字與模版數字的相似性參數

w = 要辨識的未知數字， $w = 1$ to 8

j = 單一號碼所分割的區塊， $j = 1$ to 40

i = 發票模版的號碼， $i = 0$ to 9

ω = 經過模版比對後所得到的比較參數值

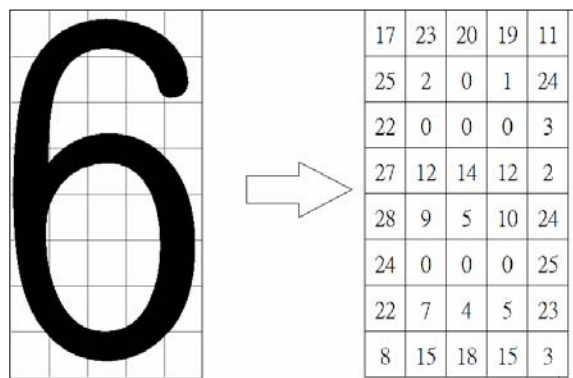


圖 6 待辨識數字的區域特徵值

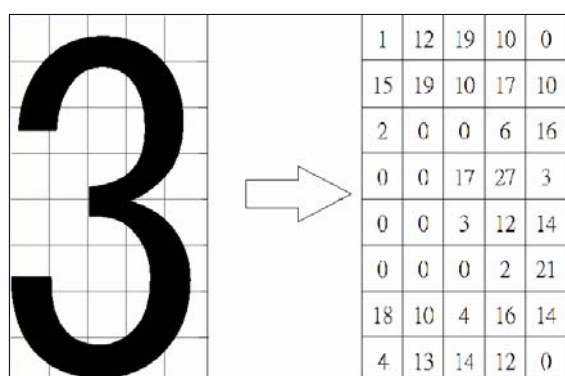


圖 7 數字 3 的區域特徵值

表 1 數字 3 的樣版區域加權值

1	1	1	1	10
1	1	1	1	5
1.5	1	1	1	1
15	1	1	1	1
15	1	1	1	1
1.5	1	1	1	1
1	1	1	1	1
1	1	1	1	10

表 2 待辨識數字之模版比較結果

待辨識的圖像 6					
模版	與0之比較	與1之比較	與2之比較	與3之比較	與4之比較
相似參數	9.95	32.00	16.76	32.75	13.88
模版	與5之比較	與6之比較	與7之比較	與8之比較	與9之比較
相似參數	17.26	8.26	74.08	9.01	11.86

表 2 是待辨識數字（在此為‘6’）與模版數字 0~9 的比對結果，由此可知，此待辨識數字為‘6’。依序將發票的 8 個未知數字作比對後由表 3 結果可知我們可以判定發票的號碼是‘64427217’。[3]

表 3 模版比對的結果

	未知數1	未知數2	未知數3	未知數4
與0比較	9.95	22.38	19.65	9.95
與1比較	32.00	35.13	31.65	33.60
與2比較	16.76	13.90	12.63	5.24
與3比較	32.75	12.31	10.79	14.10
與4比較	13.88	4.55	4.93	13.48
與5比較	17.26	25.46	23.55	11.76
與6比較	8.26	21.06	19.90	18.26
與7比較	74.08	37.83	33.50	23.28
與8比較	9.01	12.63	13.48	20.45
與9比較	11.86	20.63	18.18	10.24
	未知數5	未知數6	未知數7	未知數8
與0比較	14.40	11.03	18.35	14.93
與1比較	29.25	35.43	2.05	36.48
與2比較	9.05	5.06	8.53	9.24
與3比較	12.13	14.58	9.68	13.28
與4比較	9.68	13.45	10.23	10.30
與5比較	12.71	12.69	13.31	13.41
與6比較	18.34	18.54	13.91	20.89
與7比較	3.75	23.13	12.50	3.58
與8比較	20.85	20.98	21.40	20.98
與9比較	11.83	9.96	12.56	11.75

在年月份部份如圖 8 所示，我們還可以再縮小範圍，利用垂直投影及定位切割我們可以切出以下幾個小項如圖 9。

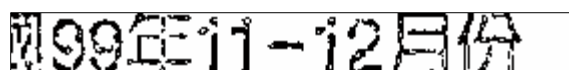


圖 8 再次裁切年月份



圖 9 年月份細分

接著再進行模版比對即可知道年月份了。

4. 模擬結果與討論

本系統所使用的軟硬設備條件如下

軟體：

Matlab 7.6

硬體：

CPU：Intel® Core™2 Duo E6300

RAM：5 Giga Byte

圖像：

每張發票尺寸約是 190mm*44mm 以乾淨的發票為主，部份含有歪斜及汙損。

使用之閾值：

年月份： $[0.85 \leq \text{亮度值}(V) \leq 0.90]$

發票號碼：

11-12 月 $[0.55 \leq \text{亮度值}(V) \leq 0.75]$ 。

9-10 月 $[0.00 \leq \text{亮度值}(V) \leq 0.55]$ 。

辨識成功的結果：如圖 10、11。

整體成功率如表 4。

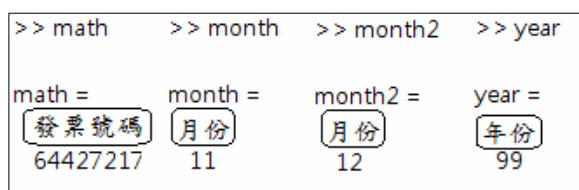


圖 10 辨識結果

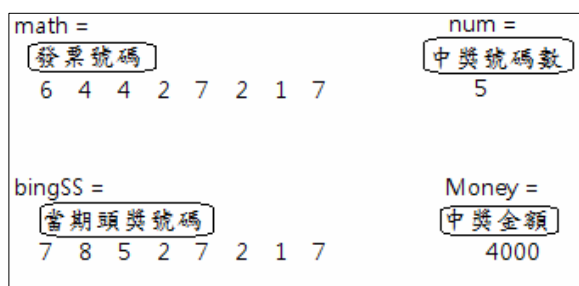


圖 11 中獎通知並顯示金額

表 4 字元切割及辨識率

項目	成功率
號碼部分投影切割	80
號碼辨識	85
年月份投影切割	84
年月份辨識	79

在本次研究當中，在投影切割方面由於會受到掃描時的傾斜的影響，傾斜角度大於 5 度在切割時就會切不出來數字的區塊，也會影響之後辨識的步驟。若投影切割成功之後則是會因為需辨識的區域受到汙損而無法將該數字及年月份正確辨識出來。

未來研究需注意的地方

1.在辨識區域的部份，畫面上的清晰程度（像是蓋上店章或是在空白區域作記號等）都會影響之後模版比對的成功率，受汙損字體在發票上會呈現模糊暈開的情況，這會使模版比對的成功率下降。2.我們可以查到說除了財政部的印刷廠之外還委託了四家的紙廠協助印刷分別是永豐紙業股份有限公司、正隆股份有限公司、大堡事務用品股份有限公司及普飛迅票證印刷股份有限公司等，所生產的發票在字型，顏色都有些許的不同，這些在不確定的點在製作模版時都需一起考慮進去並取得中間值來達到可一版多用的目的。

參考文獻

- [1] 徐曉珮，數位元影像處理，高立圖書有限公司，民國 94 年初版。
- [2] 王晉中，matlab 7 在工程上的應用，高立圖書有限公司，民國 94 年初版。
- [3] 鄭錦聰，matlab 程式設計基礎篇，全華圖書公司，民國 94 年初版。
- [4] 繆紹綱，數位元影像處理-活用 matlab，全華圖書公司，民國 88 年初版。
- [5] 繆紹綱，數位元影像處理-運用 matlab，台灣東華書局股份有限公司，民國 94 年初版。