

影像處理期末專題報告

班別：資科職碩一 學號：M11316003 姓名：何懿儒

1.1 題目：

發票號碼自動辨識兌獎系統

1.2 Introduction

1.2.1 動機

台灣的統一發票兌獎系統旨在鼓勵消費者正式索取發票，以促進稅收的正常徵收。消費者通過保留發票並參與定期的抽獎來有機會獲得獎金。目前，儘管已經有行動載具自動儲存發票的協助，但部分店家仍照常開立發票，需要消費者手動對獎，這個過程既耗時又容易出錯。因此，開發一個自動化的發票號碼辨識與對獎系統，來顯著提升使用者兌獎的便捷性與效率。

1.2.2 目的或做完此專題，要解決的問題

本系統的主要目的是利用影像處理技術自動識別發票上的號碼，並根據當月的中獎數據來進行中獎判別。此系統將應用模板匹配技術初步定位發票上的號碼區域，接著進行精確的數字識別，讀取發票的文字。期望將解決的問題包括光線不均、發票磨損或褶皺帶來的影像識別困難，最終，我們可以透過輸入當月發票中獎的號碼末三碼，將整個資料夾的所有發票進行是否中獎的判別。

1.3、文獻

1. M. D. S. (2020). "Template Matching and Its Applications in Object Detection." *Journal of Image Processing*.

內容：討論模板匹配技術基礎，分析其應用於影像檢測中的應用與限制。

2. P. Lee, et al. (2021). "Multi-Scale Template Matching for Object Recognition." *IEEE Transactions on Computer Vision*.

內容：提出了多尺度模板匹配方法，顯著提升檢測準確性。

3. L. Zhao & W. Zhang. (2022). "Real-Time Image Processing using OpenCV." *International Journal of AI*.

內容：探討了基於 OpenCV 的實時影像處理技術實現，提供工程實現參考。

1.4、我的方法

事前準備：

首先，我們使用 Anaconda Prompt 環境，執行我們的對獎系統 bill.py 檔案。並準備了 50 張不同的角度、曝光、種類的發票做為 dataset。

第一步：執行 python 檔，並讀取發票樣本

在執行 bill.py 檔案後，擬讓系統詢問當期的中獎號碼末三碼，並詢問資料及檔案夾的位置。當輸入完成後，所有發票樣本都會被一一讀取。

第二步：將發票圖像處理後，偵測擷取的内容辨識發票號碼

並將圖片轉為灰階、使用高斯模糊減少雜訊、使用 Canny 邊緣檢測找出圖片中的輪廓，最後框選出可能是發票號碼的發票內容區塊。再來，我們透過使用 Tesseract OCR 提取調整後圖片中的文字，並且定義發票號碼的規則（以英文開頭、接 8 位數字、中間可有空格或連字符），確保偵測到的是發票號碼排除其他誤判雜訊。最終用 re.findall 匹配所有可能的發票號碼，並使用 re.sub 清除不必要的勿偵測、空格與連字符。

第三部：已提取發票號碼，根據當期中獎末三碼判斷中獎與否

顯示發票號碼並判斷是否中獎，判斷最後 3 位是否與中獎號碼相同，相同輸出【中獎】，反之【沒中獎】。依此，所有 dataset 的發票樣本 50 張全數都能判斷出中獎與否，全程式截圖如下：

```
26
27 # 將圖片轉為灰階、使用高斯模糊減少雜訊、使用 Canny 邊緣檢測找出圖片中的輪廓。
28 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
29 blurred = cv2.GaussianBlur(gray, (9, 9), 0)
30 edges = cv2.Canny(blurred, 50, 300)
31
32 contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
33
34 # 將符合條件的輪廓用綠色方框標註。
35 result_image = image.copy()
36 for contour in contours:
37     x, y, w, h = cv2.boundingRect(contour)
38     if w > 10 and h > 20:
39         cv2.rectangle(result_image, (x, y), (x + w, y + h), (0, 255, 0), 2)
40
41 # 透過使用 Tesseract OCR 提取圖片中的文字
42 text = pytesseract.image_to_string(Image.open(file_path), lang='eng')
43 #print("偵測到的完整文字內容:")
44 #print(text)
45
46 # 定義發票號碼的規則（以英文開頭，後接8位數字，中間可有空格或連字符），確保偵測到的是發票號碼
47 pattern = r'[A-Z]+[-\s]*\d{8}'
48
49 # 用re.findall 匹配所有可能的發票號碼。
50 matches = re.findall(pattern, text)
51
52 #使用 re.sub 清除不必要的勿偵測、空格與連字符。
53 cleaned_matches = [re.sub(r'[-\s]+', '', match) for match in matches]
```

第一步內容之程式截圖

```

26
27 # 將圖片轉為灰階、使用高斯模糊減少雜訊、使用 Canny 邊緣檢測找出圖片中的輪廓。
28 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
29 blurred = cv2.GaussianBlur(gray, (9, 9), 0)
30 edges = cv2.Canny(blurred, 50, 300)
31
32 contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
33
34 # 將符合條件的輪廓用綠色方框標註。
35 result_image = image.copy()
36 for contour in contours:
37     x, y, w, h = cv2.boundingRect(contour)
38     if w > 10 and h > 20:
39         cv2.rectangle(result_image, (x, y), (x + w, y + h), (0, 255, 0), 2)
40
41 # 透過使用 Tesseract OCR 提取圖片中的文字
42 text = pytesseract.image_to_string(Image.open(file_path), lang='eng')
43 #print("偵測到的完整文字內容:")
44 #print(text)
45
46 # 定義發票號碼的規則 (以英文開頭, 後接8位數字, 中間可有空格或連字符), 確保偵測到的是發票號碼
47 pattern = r'[A-Z]+[-\s]*\d{8}'
48
49 # 用 re.findall 匹配所有可能的發票號碼。
50 matches = re.findall(pattern, text)
51
52 # 使用 re.sub 清除不必要的勿偵測、空格與連字符。
53 cleaned_matches = [re.sub(r'[-\s]+', '', match) for match in matches]

```

第二步內容之程式截圖

```

54
55 # 顯示發票號碼並判斷是否中獎, 判斷最後3位是否與中獎號碼相同, 輸出【中獎】或【沒中獎】
56 if cleaned_matches:
57     print("偵測到的發票號碼:")
58     for match in cleaned_matches:
59         print(match)
60         # 判斷是否中獎
61         if match[-3:] == winning_last_3_digits:
62             print("【中獎】")
63         else:
64             print("【沒中獎】")
65     else:
66         print("未能找到發票號碼。")
67
68 # 顯示處理結果
69 cv2.imshow("Processed Image", result_image)
70 cv2.waitKey(0)
71 cv2.destroyAllWindows()
72
73 # 主程式的入口, 提示使用者輸入資料夾路徑, 檢查路徑是否有效, 若有效則執行圖片處理。
74 if __name__ == "__main__":
75     folder_path = input("請輸入圖片資料夾的路徑:")
76     if os.path.isdir(folder_path):
77         process_images(folder_path)
78     else:
79         print("無效的資料夾路徑。")
80

```

第三步內容之程式截圖

1.5、實驗結果

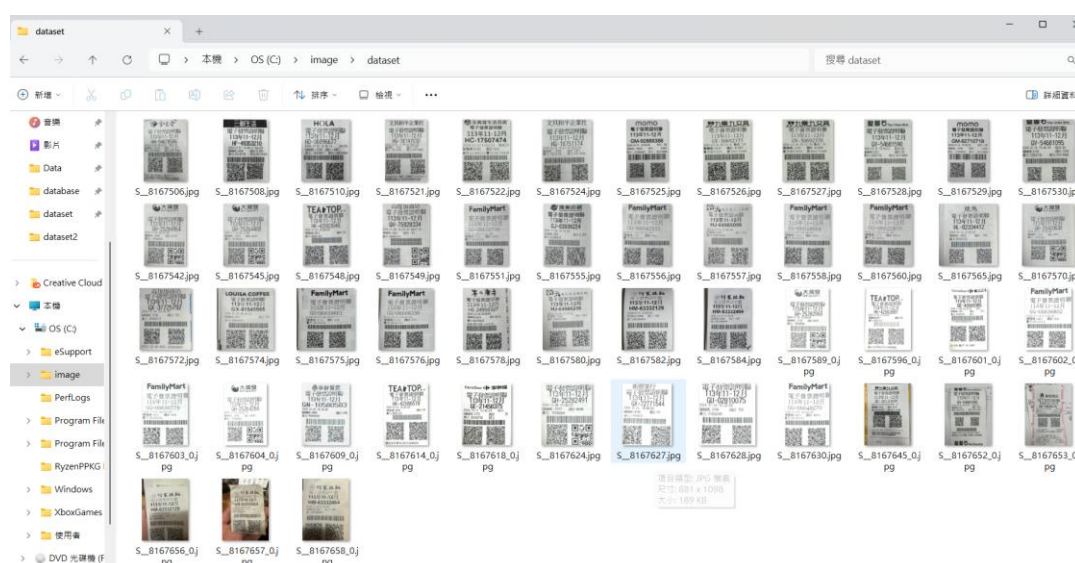
```
系統管理員: Anaconda Prompt - python bill.py

(base) C:\Windows\System32>conda activate image

(image) C:\Windows\System32>cd C:\image

(image) C:\image>python bill.py
請輸入當期的中獎號碼(3碼數字):
```

首先，我們執行 Anaconda Prompt 並以 `conda activate image` 來導向我們適合執行程式的環境，並且 `cd` 到我們專案資料夾的位置。



我們在專案資料夾內，準備一路徑為 `C:\image\dataset` 的資料夾來放置 50 張不同的 11-12 月發票樣本。以利之後的發票辨識讀取。

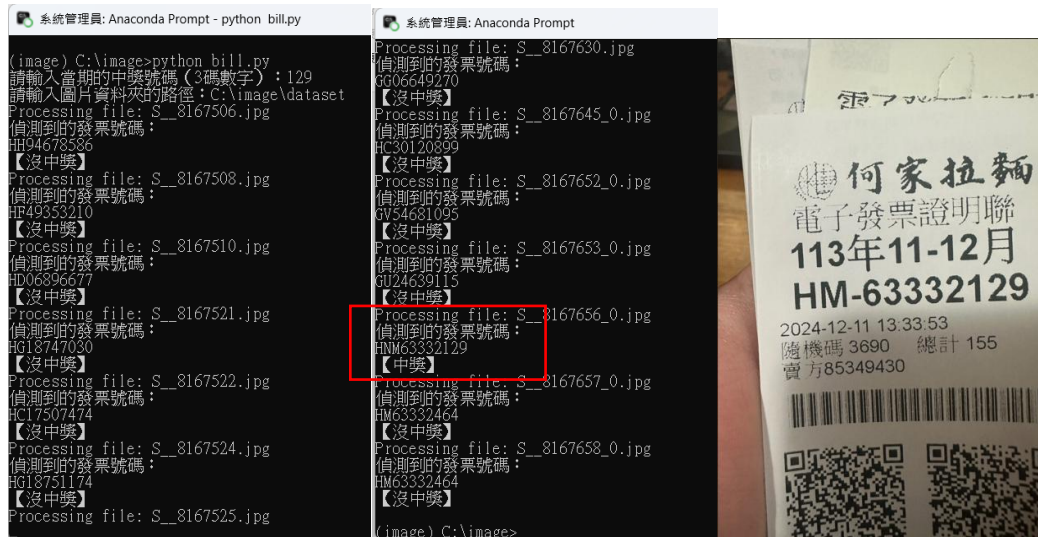
```
系統管理員: Anaconda Prompt - python bill.py

(base) C:\Windows\System32>conda activate image

(image) C:\Windows\System32>cd C:\image

(image) C:\image>python bill.py
請輸入當期的中獎號碼(3碼數字): 129
請輸入圖片資料夾的路徑: C:\image\dataset
```

在此，我們假定這些 11-12 月發票的中獎號碼末三碼為 129(當然，這 50 張樣本中，有一張是何家拉麵發票號碼為尾端 129 的，以供我們確認是否能從 50 張中判斷這張有中獎)。給予我們的圖片資料夾路徑後，讓他執行所有發票的影像辨識。



我們可以看到他開始將所有圖片一一進行兌獎，並且成功將何家拉麵的發票判斷為中獎。其他全部末三碼並非 129 的皆為沒中獎。

1.6、參考資料

1. OpenCV Official Documentation: <https://docs.opencv.org>
2. Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing (4th Edition)*. Pearson.
3. Goodfellow, I., et al. (2016). *Deep Learning*. MIT Press.