# 驗證碼 (captcha) 的光學文字辨識

- 實現網路爬蟲過程中，會遇到有些網站需要你輸入驗證碼
  - Completely Automated Public Turing test to tell Computers and Humans Apart，全自動區分電腦和人類的公開圖靈測試
- 使用 tesseract OCR(光學文字辨識 ) 獲得驗證碼
  - 透過容器： docker run --rm -v $(pwd):/data -w /data vimagick/tesseract test.png stdout -l eng --psm 13
    - 在容器內替瀏覽器安裝中文字型 apt-get install fonts-arphic-ukai fonts-arphic-uming
  - 直接將工具安裝在作業系統上： apt-get install tesseract-ocr ; apt-get install imagemagick
- 測試結果
  - 簡單的 4 位數字小圖片 (國家森林遊樂區網路訂房系統 )：八成以上成功率

# 驗證碼（國家森林遊樂區網路訂房）


待辨識和輸入的驗證碼

# 測試樣本


3692


9477


3513


3262


9198


9898


2571


0288


0534


9587


4520

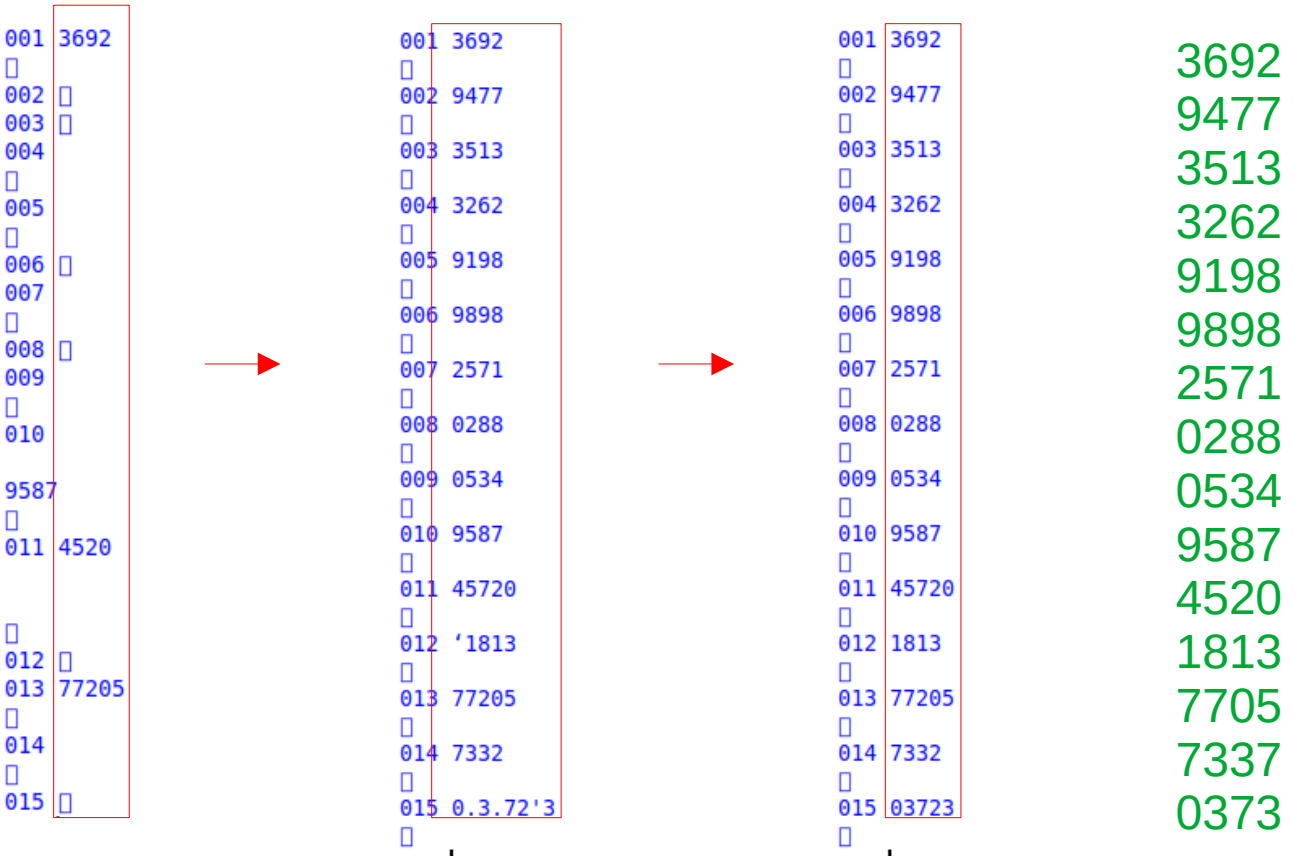

1813


7705


7337


0373

# 光學文字辨識後的結果




3692
9477
3513
3262
9198
9898
2571
0288
0534
9587
4520
1813
7705
7337
0373

最初範例的設定　　將 psm 改為 13　　限定只辨識數字　　正解

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3692 | | | | | | | | | 9587 | 4520 | | 77205 | | |
| 2 | | | | | | | | | | | | | | | |
| 3 | 3692 | | | | | | | | | 9587 | 4520 | | 77205 | | |
| 4 | 3692 | | | | | | | | | 9587 | | | 77205 | | 03723 |
| 5 | 2<br>9<br>46<br>3 | 7<br>7<br>4<br>9 | 9<br>1<br>5<br>3 | 42<br>6<br>2<br>3 | 8<br>9<br>1<br>9 | 0<br>9<br>8<br>9 | 1<br>7<br>2 | 2<br>8<br>6<br>0 | 4<br>3<br>5<br>0 | 7<br>8<br>0<br>9 | 0<br>2<br>0<br>4 | 3<br>1<br>8<br>1 | 5<br>0<br>7<br>7 | 7<br>0<br>3<br>7 | 3<br>7<br>3<br>3 |
| 6 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 4520 | 1813 | 77205 | 7332 | 03723 |
| 7 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 4520 | 1813 | 77205 | 7332 | 03723 |
| 8 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 45720 | 1813 | 77205 | 7332 | 03723 |
| 9 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 45720 | 1813 | 77205 | 7332 | 03723 |
| 10 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 4520 | 1813 | 77205 | 7332 | 03723 |
| 11 | 3692 | 9477 | 3513 | 62 | 91 | 9898 | 2571 | 0288 | 0534 | 9587 | 452 | 1813 | 77205 | 733 | 03723 |
| 12 | 3692 | 9477 | 3513 | 62 | 91 | 9898 | 71 | 0288 | 05 | 9587 | 452 | 1813 | 77205 | 733 | 0373 |
| 13 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 45720 | 1813 | 77205 | 7332 | 03723 |
| 14 | 3692 | 9477 | 3513 | 3262 | 9198 | 9898 | 2571 | 0288 | 0534 | 9587 | 4520 | 1813 | 7705 | 7337 | 0373 |

對 15 個樣本，藉由改變 psm，操作在 13 種不同模式下，最後共得到 195 辨識結果，其中以模式 6 和 7 效果最佳，模式 8,9,10 和 13 次之，而模式 2 對數字辨識明顯不行。

A ~ O : 15 samples
1 ~ 13: 13 modes
14 : 正解

← 正確數字 (row # 14)

# tesseract 簡介

https://zh.wikipedia.org/wiki/Tesseract

## Tesseract [編輯]

維基百科，自由的百科全書

此條目**可參照英語維基百科相應條目來擴充**。*(2018年7月24日)*

若您熟悉來源語言和主題，請協助參考外語維基百科擴充條目。請勿直接提交機械翻譯，也不要翻譯不可靠、低品質內容。依版權協議，譯文需在編輯摘要註明來源，或於討論頁頂部標記 `{{Translated page}}` 標籤。

**Tesseract**是一個光學字元辨識引擎，支援多種作業系統。[1]Tesseract是基於Apache許可證的自由軟體[2]，自2006 年起由Google贊助開發[3]。

2006年，Tesseract被認為是最精準的開源光學字元辨識引擎之一。[2][4]

# 透過容器來執行 tesseract

https://registry.hub.docker.com/r/vimagick/tesseract/ 100%

## tesseract

Tesseract is an Open Source OCR engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API. It supports a wide variety of languages.

Tesseract doesn't have a built-in GUI, but there are several available from the 3rdParty page.

### Quick Start

```
$ alias tesseract='docker run --rm -v `pwd`:/data -w /data vimagick/tesseract'

$ tesseract input.png output -l eng --psm 3
$ cat output.txt
The (quick) [brown] {fox} jumps!

$ tesseract chinese.jpg stdout -l chi_tra --psm 8 --oem 0
學習
```

Docker Pull Command

```
docker pull vimagick/tesseract
```

感謝該作者的啟發

Owner

vimagick

Source Repository

Github
vimagick/dockerfiles

docker run --rm -v $(pwd):/data -w /data vimagick/tesseract input.png output -l eng --psm 3

# 測試程式（1)

```
import os
#import subprocess

#docker_cmd1="docker run --rm -v $(pwd):/data -w /data vimagick/tesseract \
#test_{:03}.png stdout -l eng --psm {}"

#docker_cmd2="docker run --rm -v $(pwd):/data -w /data vimagick/tesseract \
#test_{:03}.png stdout --oem 0 -c tessedit_char_whitelist=0123456789 --psm {}"

docker_cmd3="docker run --rm -v $(pwd):/data -w /data vimagick/tesseract \
test_{:03}.png stdout -c tessedit_char_whitelist=0123456789 --psm {}"

mode = 13# 3, '1st':6,10 '2nd':=7,8,9,13 '3rd':11,12

for number in range(15):

    ocr_result = os.popen(docker_cmd3.format(number+1, mode)).read()

    print('{:03}'.format(number+1), ocr_result)
```

# 測試程式 (2)

```python
import os
import csv
import time

docker_cmd3="docker run --rm -v $(pwd):/data -w /data vimagick/tesseract \
test_{:03}.png stdout -c tessedit_char_whitelist=0123456789 --psm {}"

tmp = str(time.time_ns()) # used as unique number
mode_max = 14
sample_max = 16

for mode in range(1, mode_max):

    psm = dict()

    for sample in range(1, sample_max):

        ocr_result = os.popen(docker_cmd3.format(sample, mode)).read()

        psm_key = '{:03}'.format(sample)
        print(psm_key, ocr_result)
        psm[psm_key] = ocr_result

    fieldnames = ['{:03}'.format(sample) for sample in range(1, sample_max)]

    with open('ocr_' + tmp + '.csv', 'a', newline='', encoding='utf-8') as f_w:
        writer = csv.DictWriter(f_w, fieldnames=fieldnames)
        writer.writerow(psm)
```