

嵌入式系統軟體設計與實作 實驗報告十二

資工三 許耘熙 411410054

實驗名稱：

車牌辨識專題

實驗目的：

透過實驗模組建構簡易車牌辨識系統

實驗步驟：

1. 使用 ssh 進行遠端連線

```
ssh -X yunhsihsu@172.16.1.53
```

2. 使用 sftp 將檔案傳到樹梅派

```
sftp yunhsihsu@172.16.1.53  
put -r ./plate_recognition
```

3. 設置相關環境以及安裝套件

```
sudo -s  
apt update  
apt upgrade  
apt install autoconf automake libtool -y  
apt install build-essential cmake git libgtk2.0-dev pkg-config \\\nlibavcodec-dev libavformat-dev libswscale-dev -y  
apt install libatlas-base-dev gfortran -y  
fallocate -l 4G /swapfile  
chmod 600 /swapfile  
mkswap /swapfile  
swapon /swapfile  
swapon --show  
cd  
git clone https://github.com/DanBloomberg/leptonica.git  
cd leptonica/  
git checkout v1.71  
chmod 777 configure  
./configure  
make -j2
```

```
make install
cd
git clone https://github.com/tesseract-ocr/tesseract.git
cd tesseract/
git checkout 3.04.01
./autogen.sh
./configure --enable-debug
make -j2
make install
apt install libopencv-dev
cd
git clone https://github.com/openalpr/openalpr.git
cd openalpr/
apt install libcurl4-openssl-dev liblog4cplus-dev -y
cd src/
mkdir build
cd build
cmake ..
make -j2
make install
ldconfig
```

4. 使用 Openalpr 進行車牌辨識

```
cd ~/plate_recognition/img
alpr plate2.jpg
```

5. 完成 plate_recog.py 並執行

```
cd ~/plate_recognition
source ../env/bin/activate
pip install opencv-python
python plate_recog.py
```

6. 測量空間

```
cd ~/plate_recognition
du -sh env/ plate_recog.py template/ img/
```

問題與討論：

1. Openalpr 執行結果

```
yunhsihsu@raspberrypi: ~  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition# alpr img/plate2.jpg  
plate0: 10 results  
- MUY5686 confidence: 89.9899  
- HUY5686 confidence: 83.9382  
- MUY568 confidence: 82.9243  
- NUY5686 confidence: 82.8943  
- MUY56B6 confidence: 82.5521  
- UY5686 confidence: 81.1681  
- MUYS686 confidence: 80.0527  
- KUY5686 confidence: 79.9711  
- BUY5686 confidence: 79.9286  
- MDY5686 confidence: 79.178  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition#
```

2. plate_recog.py 執行結果



```
yunhsihsu@raspberrypi: ~  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition# python ./plate_recog.py  
y  
best match: A, value: 0.8525890111923218  
best match: A, value: 0.8694927096366882  
best match: A, value: 0.872961699962616  
best match: 4, value: 0.35498046875  
best match: S, value: 0.4684656858444214  
best match: 8, value: 0.7091379165649414  
best match: 5, value: 0.7426824569702148  
plate number: AAA4S85  
Total time: 0.209 s  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition#
```

3. 時間與空間的測量

```
yunhsihsu@raspberrypi: ~  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition# du -sh ../env/ plate_recog.py plate_recog_more.py ./img/ ./template/  
287M    ../env/  
8.0K    plate_recog.py  
8.0K    plate_recog_more.py  
4.7M    ./img/  
148K    ./template/  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition#
```

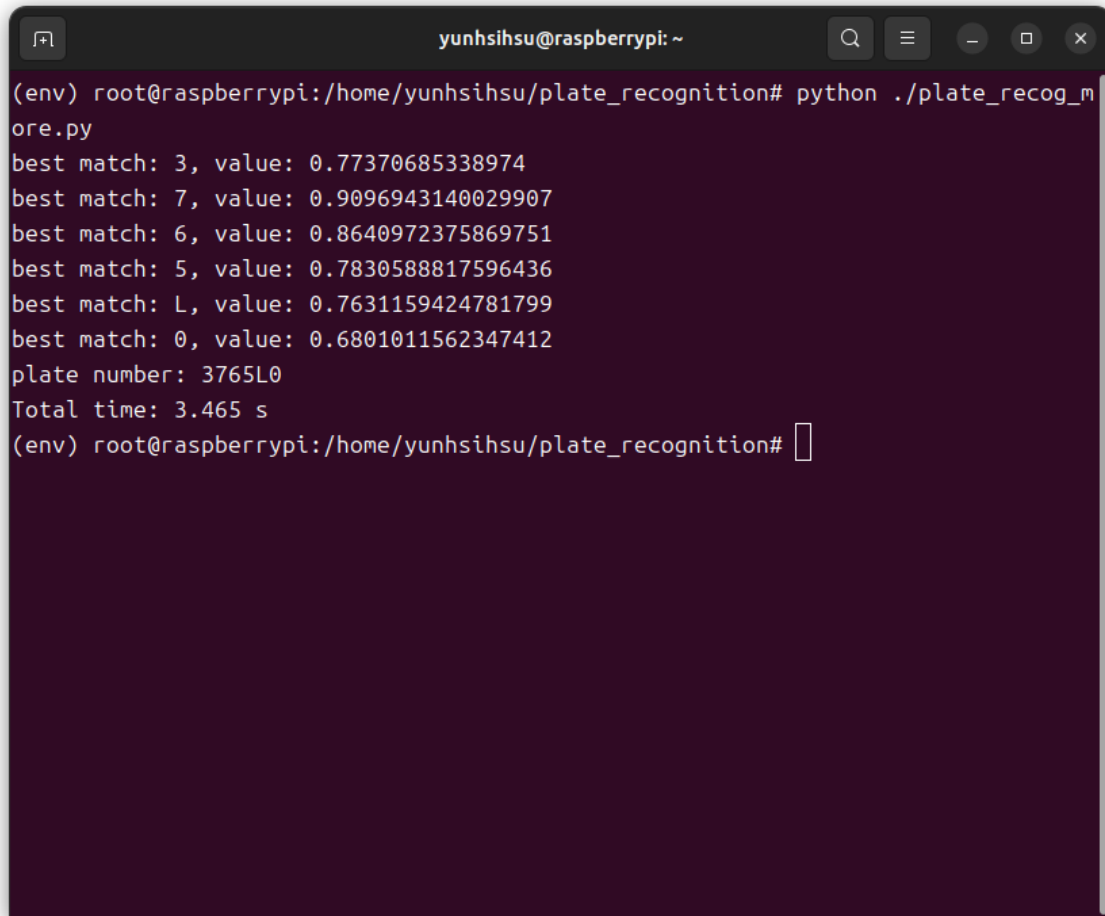
時間測量有附在執行結果中

4. 三個可能的問題點：

- (1)由於程式碼中 `num_of_digits` 被寫為 7，因此並無法辨識舊式的 6 碼車牌
- (2)沒有套入 7 碼車牌規則，造成英文與數字在不正確的地方被誤判
- (3)光線不足時可能會無法正確獲得正確的車牌資訊

5. 問題改善

撰寫 `plate_recog_more.py` 程式碼並執行



```
yunhsihsu@raspberrypi: ~  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition# python ./plate_recog_more.py  
best match: 3, value: 0.77370685338974  
best match: 7, value: 0.9096943140029907  
best match: 6, value: 0.8640972375869751  
best match: 5, value: 0.7830588817596436  
best match: L, value: 0.7631159424781799  
best match: 0, value: 0.6801011562347412  
plate number: 3765L0  
Total time: 3.465 s  
(env) root@raspberrypi:/home/yunhsihsu/plate_recognition#
```

可以測量到舊式的 6 碼車牌

6. 實驗心得

在本次實驗中，我了解到了如何使用 `OpenCV` 套件實作車牌辨識，又再次強化了我對電腦視覺影像處理技術的理解，在本實驗的延伸應用方面有遇到了不小的困難，為了實現不同形式的車牌，需要採用自動判斷位數的方法，後來為了解決自己拍的车牌無法識別，又新增了出裡光線不良的方法，最後才完成，耽誤了好多時間。