

第30章

建立哈爾特徵分類器-車牌辨 識

30-1：準備正樣本與負樣本影像資料

- 這一章的內容是要建立可以辨識汽車車牌的哈爾(Haar) 特徵分類器資源檔案，這時必須準備2類影像，其中含汽車車牌的影像我們可以稱正(Positive)樣本影像，不含汽車車牌的影像稱負(Negative)樣本影像。

30-11：準備正樣本影像 - 含汽車車牌影像



- 許多車牌辨識的停車場會將攝影機架在車身引擎蓋相同的高度或是略高一點，所以我們在拍攝汽車影像時，最好也是如此，建議拍攝汽車影像時注意下列三點：
- 1：固定高度。
- 2：固定距離。
- 3：光線良好，可以清楚顯示車身與車牌。

30-1-2：準備負樣本影像 – 不含汽車車牌影像

- 所謂的**負樣本影像**就是指不含汽車車牌的影像，由於我們要訓練電腦可以認識汽車車牌，所以要準備一系列不含汽車車牌的影像告訴系統這些影像是不含汽車車牌，這些影像最好是包羅萬象，越多越好，本書筆者準備了約**295**張影像。
- 建議讀者學會本章內容後，可以準備**1000**張以上的影像。

30-2：處理正樣本影像



30-21：將正樣本影像處理成固定寬度與高度

- 程式實例 `ch30_1.py`：將所有在 `ch30/srcCar` 資料夾內的檔案，處理成寬與高分別是 **320** 和 **240** 像素的影像，然後儲存在 `ch30/dstCar` 資料夾。因為程式會有多次測試，所以這個程式第 **12** – **15** 列筆者會先刪除原資料夾內的影像。

```
===== RESTART: D:\OpenCV_Python\ch30\ch30_1.py =====  
執行 srcCar 資料夾內尺寸的轉換 ...  
儲存 dstCar 資料夾內尺寸的轉換 ...
```

Data (D:) > OpenCV_Python > ch30 > dstCar

+jQuery



car1



car2



car3

30-22：將正樣本影像轉成bmp檔案

- 程式實例ch30_2.py：由於要將dstCar資料夾所有汽車由jpg檔案轉成bmp檔案，我們必須要讀取所有dstCar資料夾的檔案，這時需要使用glob模組的glob()函數，這個函數可以得到檔案串列，這個實例是將檔案串列的路徑與檔案名稱拆開。

```
===== RESTART: D:\OpenCV_Python\ch30\ch30_2.py =====  
目前資料夾檔案名稱 =  
['dstCar\\car1.jpg', 'dstCar\\car10.jpg', 'dstCar\\car11.jpg', 'dstCar\\car12.jp  
g', 'dstCar\\car13.jpg', 'dstCar\\car14.jpg', 'dstCar\\car15.jpg', 'dstCar\\car1  
6.jpg', 'dstCar\\car17.jpg', 'dstCar\\car18.jpg', 'dstCar\\car19.jpg', 'dstCar\\  
car2.jpg', 'dstCar\\car20.jpg', 'dstCar\\car21.jpg', 'dstCar\\car22.jpg', 'dstCa  
r\\car23.jpg', 'dstCar\\car24.jpg', 'dstCar\\car25.jpg', 'dstCar\\car26.jpg', 'd  
stCar\\car27.jpg', 'dstCar\\car28.jpg', 'dstCar\\car29.jpg', 'dstCar\\car3.jpg',  
 'dstCar\\car30.jpg', 'dstCar\\car31.jpg', 'dstCar\\car32.jpg', 'dstCar\\car33.j  
pg', 'dstCar\\car34.jpg', 'dstCar\\car35.jpg', 'dstCar\\car36.jpg', 'dstCar\\car  
37.jpg', 'dstCar\\car38.jpg', 'dstCar\\car39.jpg', 'dstCar\\car4.jpg', 'dstCar\\  
car40.jpg', 'dstCar\\car41.jpg', 'dstCar\\car42.jpg', 'dstCar\\car43.jpg', 'dstC  
ar\\car44.jpg', 'dstCar\\car45.jpg', 'dstCar\\car46.jpg', 'dstCar\\car47.jpg', '  
dstCar\\car48.jpg', 'dstCar\\car49.jpg', 'dstCar\\car5.jpg', 'dstCar\\car50.jpg',  
 'dstCar\\car51.jpg', 'dstCar\\car52.jpg', 'dstCar\\car53.jpg', 'dstCar\\car54.  
jpg', 'dstCar\\car55.jpg', 'dstCar\\car56.jpg', 'dstCar\\car57.jpg', 'dstCar\\ca  
r58.jpg', 'dstCar\\car59.jpg', 'dstCar\\car6.jpg', 'dstCar\\car60.jpg', 'dstCar\\  
\\car61.jpg', 'dstCar\\car62.jpg', 'dstCar\\car63.jpg', 'dstCar\\car64.jpg', 'dst  
Car\\car65.jpg', 'dstCar\\car66.jpg', 'dstCar\\car67.jpg', 'dstCar\\car68.jpg',  
 'dstCar\\car69.jpg', 'dstCar\\car7.jpg', 'dstCar\\car70.jpg', 'dstCar\\car71.jpg  
, 'dstCar\\car72.jpg', 'dstCar\\car73.jpg', 'dstCar\\car74.jpg', 'dstCar\\car75  
.jpg', 'dstCar\\car76.jpg', 'dstCar\\car77.jpg', 'dstCar\\car78.jpg', 'dstCar\\c  
ar79.jpg', 'dstCar\\car8.jpg', 'dstCar\\car80.jpg', 'dstCar\\car81.jpg', 'dstCar  
\\car82.jpg', 'dstCar\\car83.jpg', 'dstCar\\car84.jpg', 'dstCar\\car85.jpg', 'ds  
tCar\\car86.jpg', 'dstCar\\car87.jpg', 'dstCar\\car88.jpg', 'dstCar\\car89.jpg',  
 'dstCar\\car9.jpg', 'dstCar\\car90.jpg']  
['dstCar', 'car1.jpg']  
['dstCar', 'car10.jpg']  
['dstCar', 'car11.jpg']
```

- 程式實例ch30_3.py：將所有ch30/dstCar資料夾內的.jpg汽車影像轉成.bmp影像，同時存入ch30/bmpCar資料夾。

===== RESTART: D:\OpenCV_Python\ch30\ch30_3.py =====
在 bmpCar 資料夾重新命名車輛副檔名成功



30-3：處理負樣本影像

- 程式實例ch30_4.py：將ch30/notCar資料夾的所有負樣本影像轉為灰階，檔案名稱改為notcar*.jpg，其中*是檔案編號，同時將寬與高改為500和400，然後存至ch30/notCarGray資料夾。

```
===== RESTART: D:\OpenCV_Python\ch30\ch30_4.py =====  
在 notCar 資料夾將影像轉為灰階成功,同時存入notCarGray資料夾
```

Data (D:) > OpenCV_Python > ch30 > notCar



antar



antarctic



dogcat



eagle

Data (D:) > OpenCV_Python > ch30 > notCarGray



notcar1



notcar2



notcar3





notcar4

30-4：建立辨識車牌的哈爾(Haar) 特徵分類器

- 30-4-1：下載建立哈爾特徵分類器工具
- 請進入下列網址。
- <https://github.com/sauhaardac/haar-training>
- 然後可以看到下列網頁內容。


← → ↻ github.com/sauhaardac/haar-training




 Why GitHub? ▾ Team Enterprise Explore ▾ Marketplace Pricing ▾


 [sauhaardac](#) / [Haar-Training](#) Public Notifications


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

[master](#) ▾ [1 branch](#) [0 tags](#) [Go to file](#) [Code ▾](#) [About](#)


 [sauhaardac](#) Update README.md

 cascade2xml	First commit
 training	First commit
 README.md	Update README.


 README.md



 Clone ?

HTTPS GitHub CLI



Use Git or checkout with SVN using the web URL.

 Open with GitHub Desktop

 Download ZIP 

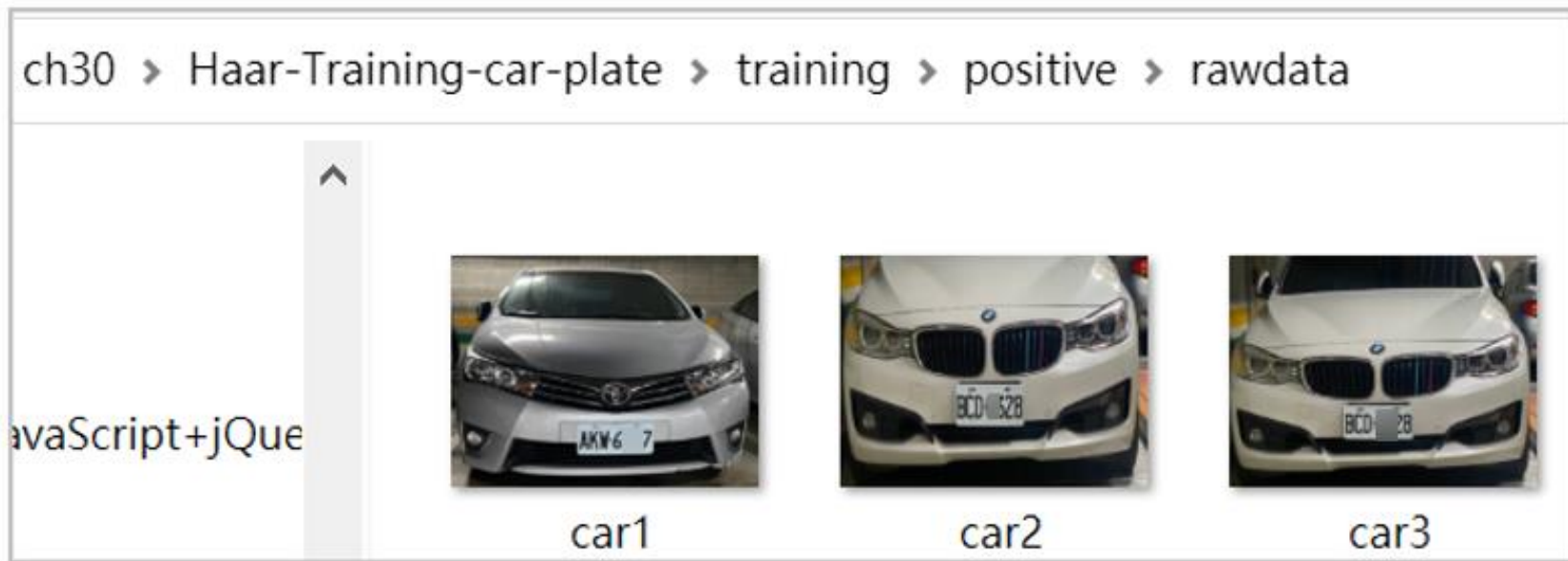
A comprehensive
Cascades of
operating s
All necessa
stored with
usage instr

[Readme](#)

Releases

30-4-2：儲存正樣本影像

- 正樣本影像是必須儲存在下列資料夾：
- [ch30/Haar-Training-car-plate/training/positive/rawdata](#)



30-4-3：儲存負樣本影像

- 負樣本影像是儲存在下列資料夾：
- `ch30/Haar-Training-car-plate/training/negative`
- 請執行下列步驟：
- 1：先將上述資料夾所有影像檔案和**bg.txt**檔案刪除。
- 2：只保留**create_list.bat**。
- 3：將程式實例**ch30_4.py**所建立的**ch30/notCarGray**資料夾內所有灰階影像複製至此資料夾。

ch30 > Haar-Training-car-plate > training > negative

	名稱	日期
	bg	2021/12/10 上午 12:53
JavaScript+jQuery	create_list	2016/10/1 上午 05:26
	notcar1	2021/12/9 下午 10:55
	notcar2	2021/12/9 下午 10:55

```
notcar1.jpg  
notcar10.jpg  
notcar100.jpg  
notcar101.jpg  
notcar102.jpg  
notcar103.jpg
```

30-4-4：為正樣本加上標記

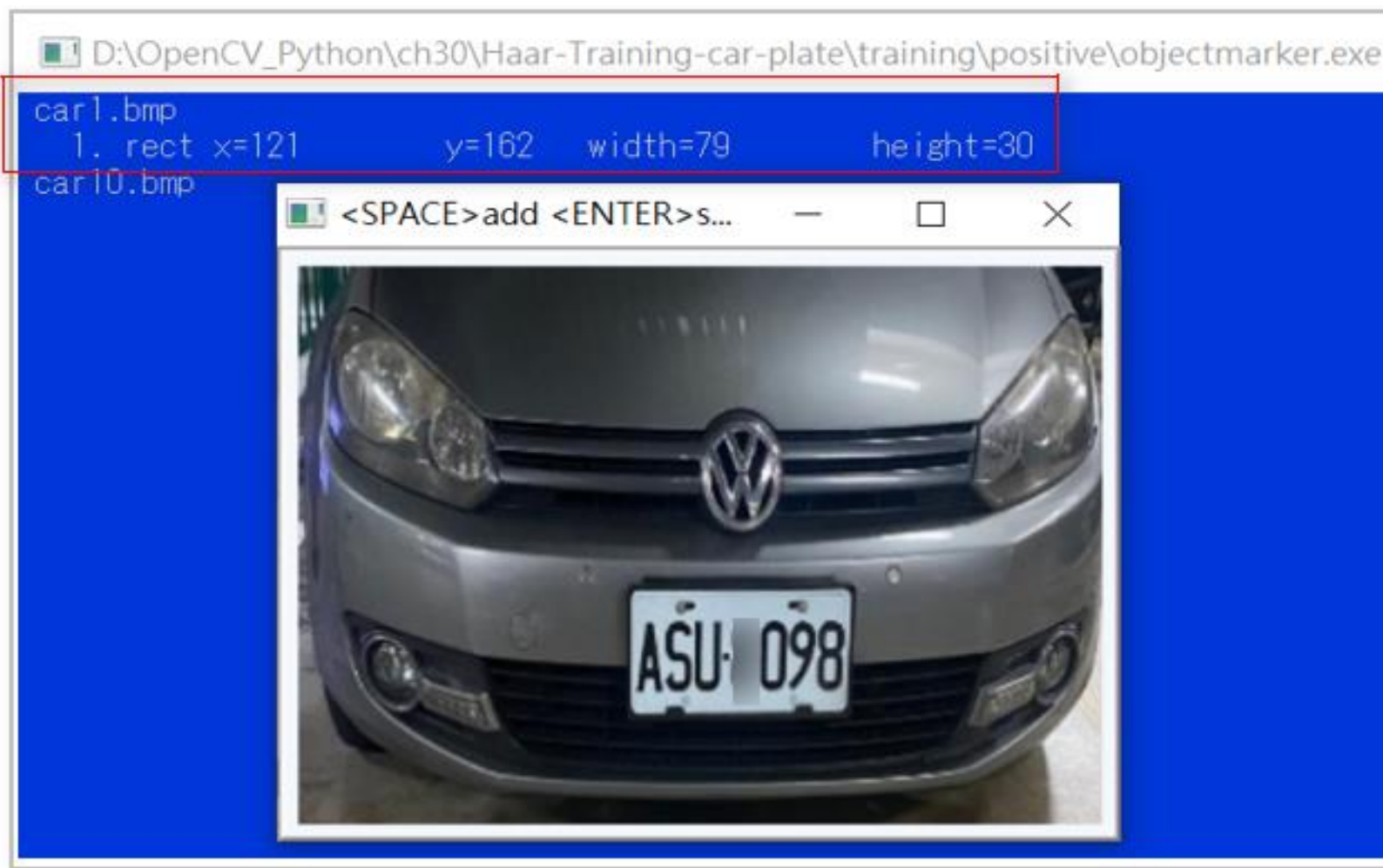
- 請開啟 [Haar-Training-car-plate/training/positive](#) 資料夾內的 [objectmarker.exe](#) 檔案，連按兩下可以開啟正的汽車樣本影像，然後請為每部車子的車牌加上外框，這個加外框的動作也稱標記，標記方式如下：
 - 1：將滑鼠游標移至車牌左上角，拖曳至車牌右下角，可以建立車牌框。
 - 2：同時按空白鍵和Enter，可以自動出現下一輛車的影像。

D:\OpenCV_Python\ch30\Haar-Training-car-plate\training\positive\objectmarker.exe

car1.bmp



- 上述框選完車牌後請同時按空白鍵和**Enter**，可以看到所框選的左上角座標和**width**與**height**，同時顯示下一部車供框選。



- 標記完成後，在相同的資料夾可以看到info.txt檔案，這個檔案紀錄正樣本影像的路徑與檔名、標記數量、標記座標、寬與高。

	標記數量	寬與高			
檔案路徑與名稱		標記座標			
rawdata/car1.bmp	1	133	177	76	27
rawdata/car10.bmp	1	134	148	86	38
rawdata/car11.bmp	1	113	172	114	34
rawdata/car12.bmp	1	143	182	72	30
rawdata/car13.bmp	1	119	180	96	28

30-4-5：設計程式顯示標記

- 前一節我們為每個正樣本影像建立標記了，現在可以使用程式了解所建立的正樣本影像標記，如果感覺位置有偏差可以修訂info.txt的內容。
- 註：如果重新執行objectmarker.exe會造成原先的標記消失。
- 程式實例ch30_5.py：顯示以及繪製車牌框線，讀者可以在ch30/plate-mark資料夾看到所有框選的結果。

```
===== RESTART: D:\OpenCV_Python\ch30\ch30_5.py =====
```

```
car1.bmp  
car10.bmp  
car11.bmp  
car12.bmp  
car13.bmp
```

....

```
car88.bmp  
car89.bmp  
car9.bmp  
car90.bmp  
繪製車牌框完成
```


- 開啟ch30/plate-mark資料夾看到所有框選的結果。



30-5：訓練辨識車牌的哈爾特徵分類器

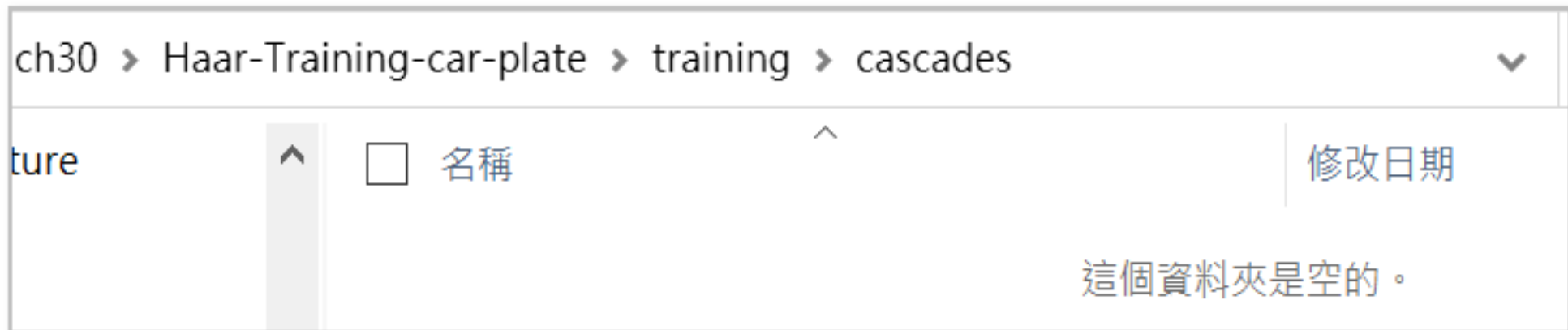
- 30-5.1：建立向量檔案
- 請參考下列修改內容：
- `createsamples.exe -info positive/info.txt -vec vector/facevector.vec`
- `-num 90 -w 70 -h 20`
- 上述內容與意義如下：
 - ❑ `createsamples.exe`：打包向量檔案的程式。
 - ❑ `info positive/info.txt`：`positive/info.txt`是正樣本標記的路徑。
 - ❑ `vec vector/facevector.vec`：建立向量檔的路徑和檔案名稱，相當於將向量檔案建立在`vector`資料夾，使用`facevector.vec`命名。
 - ❑ `num`：正樣本影像的數量。
 - ❑ `w`：偵測標記的寬度，
 - ❑ `h`：偵測標記的高度。

- 連按兩下 `samples_creation.bat` 可以在 `vector` 資料夾建立 `facevector.vec` 向量檔案，下列是執行結果。

ch30 > Haar-Training-car-plate > training > vector		▼	↺
<input type="checkbox"/>	名稱	修改日期	
<input type="checkbox"/>	facevector.vec	2021/12/10 上午 12:42	

30-5-2：訓練哈爾分類器

- 請刪除[ch30/Haar-Training-car-plate/training/cascades](#)資料夾內容，如下所示：



- 請編輯ch30/Haar-Training-car-plate/training資料夾的haartraining.bat，這是批次檔，請參考下列修改內容：

- haartraining.exe -data cascades -vec vector/facevector.vec -bg negative/bg.txt
- -npos 90 -nneg 295 -nstages 15 -mem 512 -mode ALL -w 70 -h 20 -nonsym
- 上述內容與意義如下：

- ❑ haartraining.exe：訓練哈爾分類器的執行檔程式。
- ❑ data cascades：data是指未來要儲存訓練結果的資料夾，cascades是儲存結果的資料夾。
- ❑ vec vector/facevector.vec：vector/facevector.vec是指正樣本的向量檔案。
- ❑ bg negative/bg.txt：是指負樣本的檔案。
- ❑ npos 90：是指正樣本的數量。
- ❑ nneg：是指負樣本的數量。
- ❑ nstages：是指訓練的級數，一般訓練級數越多所需時間越多，一般可以設定在12 ~ 20之間。
- ❑ mem：哈爾特徵訓練資料所需的記憶體，記憶體越大所需時間越多。
- ❑ mode：哈爾特徵的訓練模式，一般常用選項有，BASIC是指現性特徵、CORE是指線性和核心特徵、ALL是指使用所有特徵。
- ❑ w：偵測物件的寬度width。
- ❑ h：偵測物件的高度height。
- ❑ nonsym：偵測物件使用非對稱方式，如果想要使用對稱方式可以設為sym。

- 然後連點此批次檔兩下，執行此檔案，當看到下列畫面表示開始訓練資料

Number of features used : 956835

Parent node: NULL

```
*** 1 cluster ***
```

```
POS: 70 70 1.000000
```












NEG: 295 1

BACKGROUND PROCESSING TIME: 0.01

```
Precalculation time: 8.89
```

N	%SMP	F	ST.THR	HR	FA	EXP. ERR
1	100%	-	-0.942940	1.000000	1.000000	0.027397

- 訓練結束後，可以在[ch30/haar-training-car-plate/training/cascade](#)資料夾看到下列訓練結果。

Haar-Training-car-plate > training > cascades >			▼	↺		
+jQuery	^	<input type="checkbox"/> 名稱	^			修改日期
		 0				2021/12/10 上午 12:57
		 1				2021/12/10 上午 12:58
		 2				2021/12/10 上午 01:00
		 3				2021/12/10 上午 01:03
		 4				2021/12/10 上午 01:05
		 5				2021/12/10 上午 01:06
		 6				2021/12/10 上午 01:08
		 7				2021/12/10 上午 01:10
		 8				2021/12/10 上午 01:12
		 9				2021/12/10 上午 01:16
Script-		 10				2021/12/10 上午 01:18

30-5-3：建立哈爾特徵分類器資源檔

- 請編輯 `ch30/Haar-Training-car-plate/cascade2xml` 資料夾的 `convert.bat` 檔案，內容如下：
- `haarconv.exe ../training/cascades ../../haar_carplate.xml 70 20`
- 上述../代表上一層資料夾，其他內容與意義如下：
 - ❑ `haarconv.exe`：建立哈爾特徵分類器資源檔所需要的可執行檔案。
 - ❑ `../training/cascades`：這是指哈爾特徵分類器的訓練結果資料夾，可以參考30-5-2節。
 - ❑ `../../haar_carplate.xml`：這是我們要建立的哈爾特徵分類器資源檔。
 - ❑ `70 20`：偵測物件的寬度和高度。

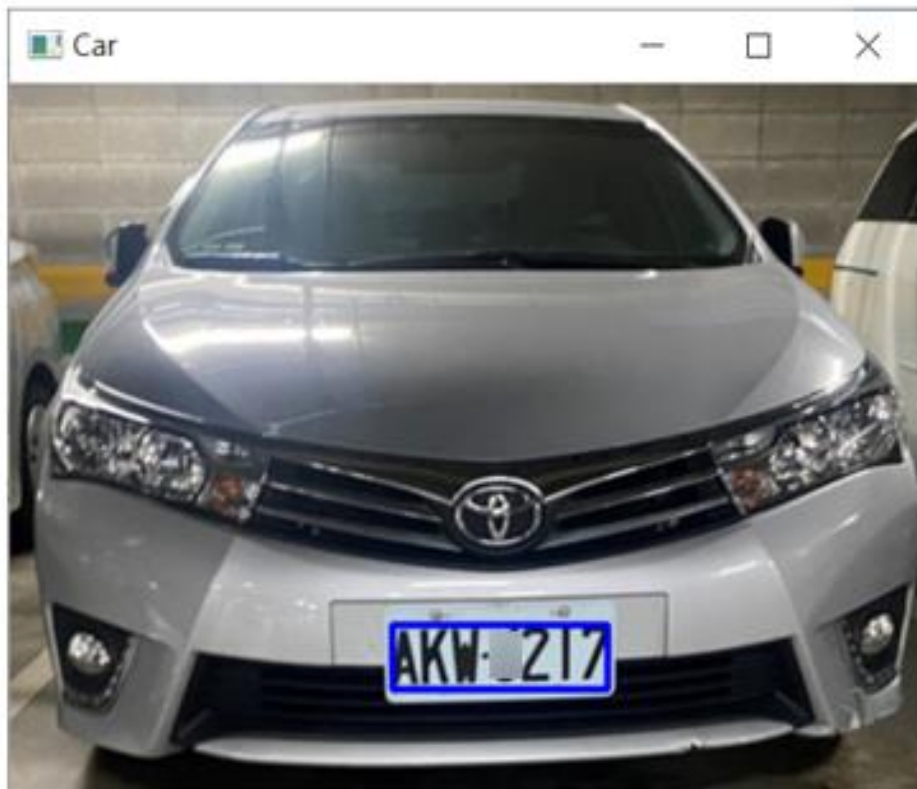
30-6：車牌偵測

- 程式實例ch30_6.py：偵測cartest1.jpg車牌的實例。

```
===== RESTART: D:/OpenCV_Python/ch30/ch30_6.py =====  
[[193 338 146  42]]
```



- 下列兩張影像分別是ch30_6_1.py和ch30_6_2.py測試cartest2.jpg和cartest3.jpg的結果。



30-7：心得報告