基於YOLOv2之汽車廠牌辨識系統

學生：0524002陳穎汶

0524004邱淳彥

0524014歐冠廷

指導老師：李嘉紘 老師

1. 研究動機與目的

**1.1 研究動機**

目前的汽車型號辨識較常應用在智能交通系統(ITS)上，車型辨識(MMR)也能夠在交通安全上有著更高的安全性，幫助警方追緝犯罪車輛。在這過去幾年也提出了非常多的技術，而基本上都是針對輪廓來進行辨識。

常見的辨識方式有，尺度不便特徵轉換(SIFT)[1]、加速魯棒特徵法(SURF)、金字塔式梯度方向直方圖(PHOG)[2]以及Gabor濾波器[3]，在某些技術中，有很多組合了低階及高階的技術功能。

車型辨識若是可以用在商務層面的話，我們希望能夠結合車廠，對購車民眾亦或是對汽車有興趣的民眾，來推出一系列的促銷活動，以及汽車間的比價資訊，同型車輛(休旅車、轎旅車、轎車等)性能間的互相比較，亦可結合網民的心得，來讓民眾能夠針對需求，取得整合性的資訊，來讓民眾購車能夠有更方便且即時的購車優惠。

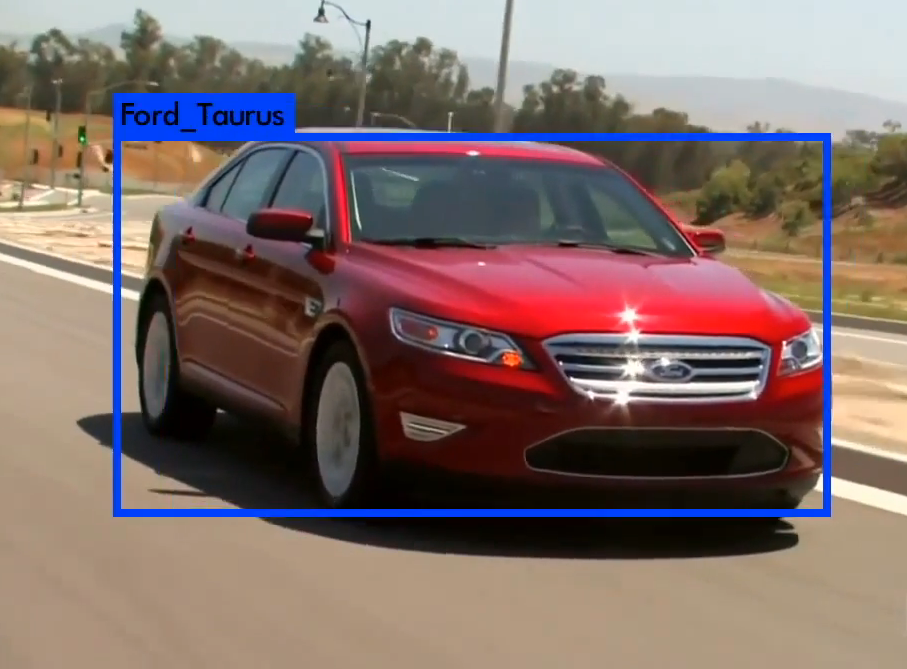


圖1、YOLOv2辨識車型

**1.2 研究目的**

本研究希望透過使用者拍攝下的影像，結合深度學習來達到影像辨識的目標。為了完成此研究，必須要有一定的樣本數量，來供學習。且如何有效地完成辨識及提高辨識率，利用YOLOv2來辨識出各廠牌LOGO以供後續型號辨識來使用，都是本研究的主要方向。

1. 文獻探討

**2.1 汽車廠牌(logo)辨識：**

Mustafa S. Kadhm[4]等人提出了一種基於OpenCV數據庫[5]的汽車廠牌logo檢測以及提取的框架，整體辨識流程如圖2。他們使用較為簡單的方式來進行辨識，步驟如下：第一是灰階化影像，原影像輸入為RGB色彩，轉為灰階目的是為了減少處理影像的時間並壓低影像的大小，第二是將影像作二值化的轉換，這裡他們使用的是OpenCV所提供的二進制閾值公式來進行轉換，目的是為了要去除影像中不必要的雜訊，第三是將影像的大小調整為原來的一半，並使用5x5的高斯模糊來將二值化後的影像降低雜訊，也是為了使汽車logo保持原有的良好形狀，第四是將影像調整回原來的大小，可以讓汽車logo有更好的表現，前面三步的前處理，會使得影像處理運行的更快且更完善，第五使用的是邊緣檢測法，用的是Canny邊緣檢測算子，第六步則是比對，透過OpenCV數據庫的模板比對函數，來找尋汽車logo的所在地，最後成功辨識了汽車的廠牌。

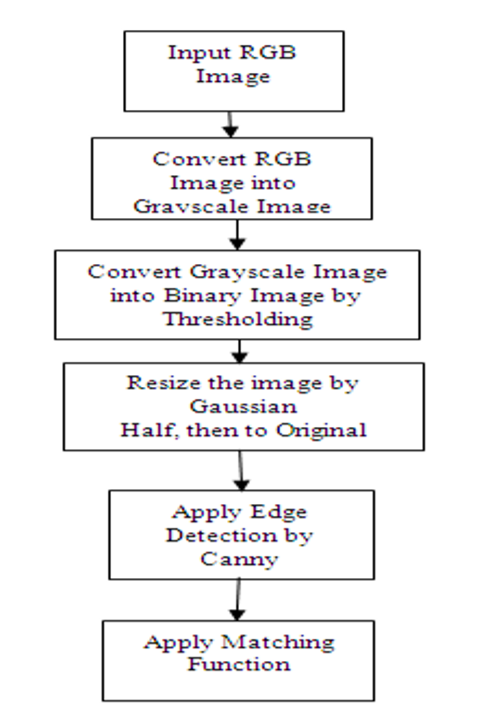


圖2、擬議框架的流程圖

中國海南大學信息科學與技術學院的張素文[6]等人，提出了一個多尺度並行卷積神經網路來辨識汽車logo，結構如圖3，而多尺度並行CNN是能以並行的方式來從原始影像中擷取特徵，主要是可以在明暗變化及影像雜訊多的情況下，仍然保有高精度辨識率的方法，而因為資料集數量不多，所以辨識率在98.80%，但在其他論文中，資料集夠多的情況下，辨識率可以高達99.80%。

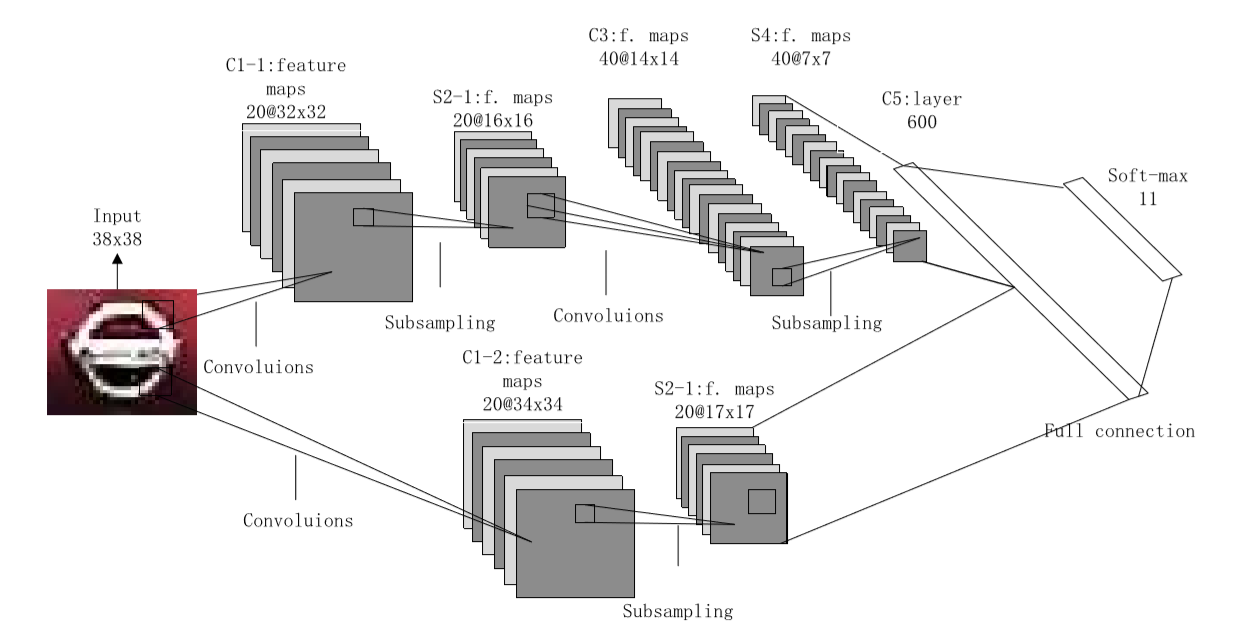


圖3、多尺度並行CNN的體系結構

**2.2 汽車型號辨識：**

在Malisa Huzaifa[7]等人的實驗中，整體流程如圖4，他們利用二進制魯棒不變的可伸縮關鍵點(Binary robust invariant scalable keypoints ,BRISK)[8]來偵測影像的關鍵點，然後結合隨機抽樣一致(RANdom SAmple Consensus ,RANSAC)[9]來降低漢明距離產生的錯誤率，在BRISK中得到的子八度和子八度尺度內層空間會應用於FAST檢測器(Features From Accelerated Segment Test ,FAST)[10]中，FAST檢測器將在16個盤旋採樣點中進行9點的測試，以識別關鍵點，得到的FAST分數將成為最大限度，最後他們的辨識成功率則是96.25%。

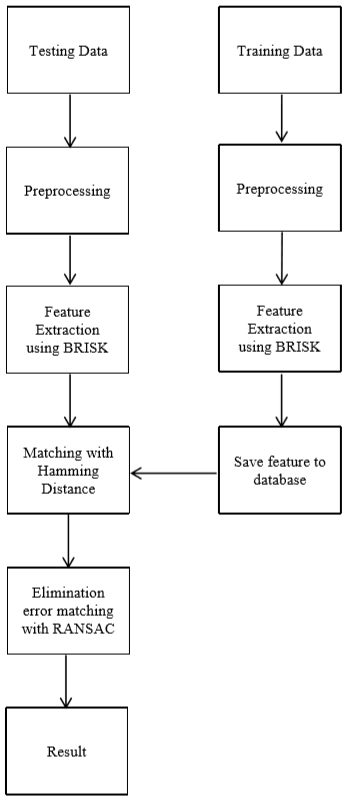


圖4、系統流程圖

**2.3 YOLOv2：**

在Noppakun Boonsim[11]等人的研究中提到，YOLOv2是基於使用卷積神經網路(Convolutional Neural Networks, CNN)一次性的物件偵測。YOLOv2利用整張影像的特徵來預測每個邊界框(bounding box)，而且還能同時預測整張影像可能出現的邊界框。

而YOLOv2本身是YOLOv1的改進版，對於其訓練及辨識的用法，都有很大的改變，像是精度上的改進、速度的改進等。

在精度上的改進：

Batch Normalization可以提升模型收斂速度，而且可以起到一定正則化效果，降低模型的過擬合(overfitting)。在YOLOv2中，每個卷積層後面都添加了Batch Normalization層，並且不再使用droput。使用Batch Normalization後，YOLOv2的mAP提升了2.4%。

而High Resolution Classifier則是在YOLOv2中，作者首先對分類網路（自訂的darknet）進行了fine tune，解析度改成448 \* 448，在ImageNet資料集上訓練10個epochs，訓練後的網路就可以適應高解析度的輸入了。然後，作者對檢測網路部分也進行fine tune。這樣通過提升輸入的解析度，mAP獲得了4%的提升。

在速度上的改進：

Darknet-19是YOLOv2採用的一個全新的特徵提取器，即為Darknet-19，它包含了19個卷積層和5個最大池化層(maxpooling)。主要採用3x3的卷積以及2x2的最大池化層之後，將特徵圖的維度降低2倍，並同時將特徵圖的通道(channel)提高2倍，因此計算量可以降低約33%。

YOLOv2在訓練上分為三種階段，如圖5，第一是在ImageNet數據集上預先訓練Darknet-19，在這一步模型輸入為224x224，總共訓練160個epochs，第二步則是將模型輸入調整為448x448，然後一樣繼續在ImageNet數據集上的finetune分類模型，訓練10個epochs，最後一步是將Darknet-19的分類模型修改為檢測模型，並在檢測數據集上繼續finetune網路。

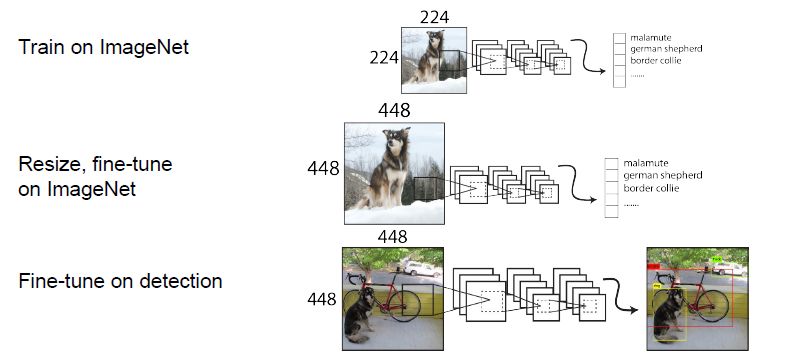


圖5、YOLOv2的訓練三階段

1. 研究方法及步驟

**3.1系統架構**

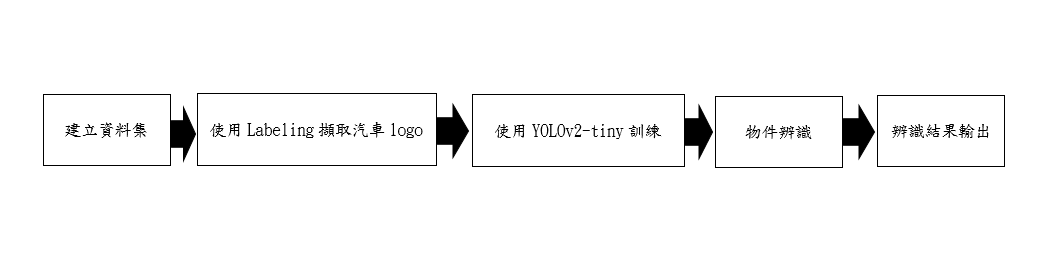


圖6、YOLOv2辨識系統架構

**3.2操作流程**

第一步：準備要訓練的資料集(如圖7)

圖7、資料集一類100張

第二步：使用labelImg將汽車logo標記起來(如圖8、9)

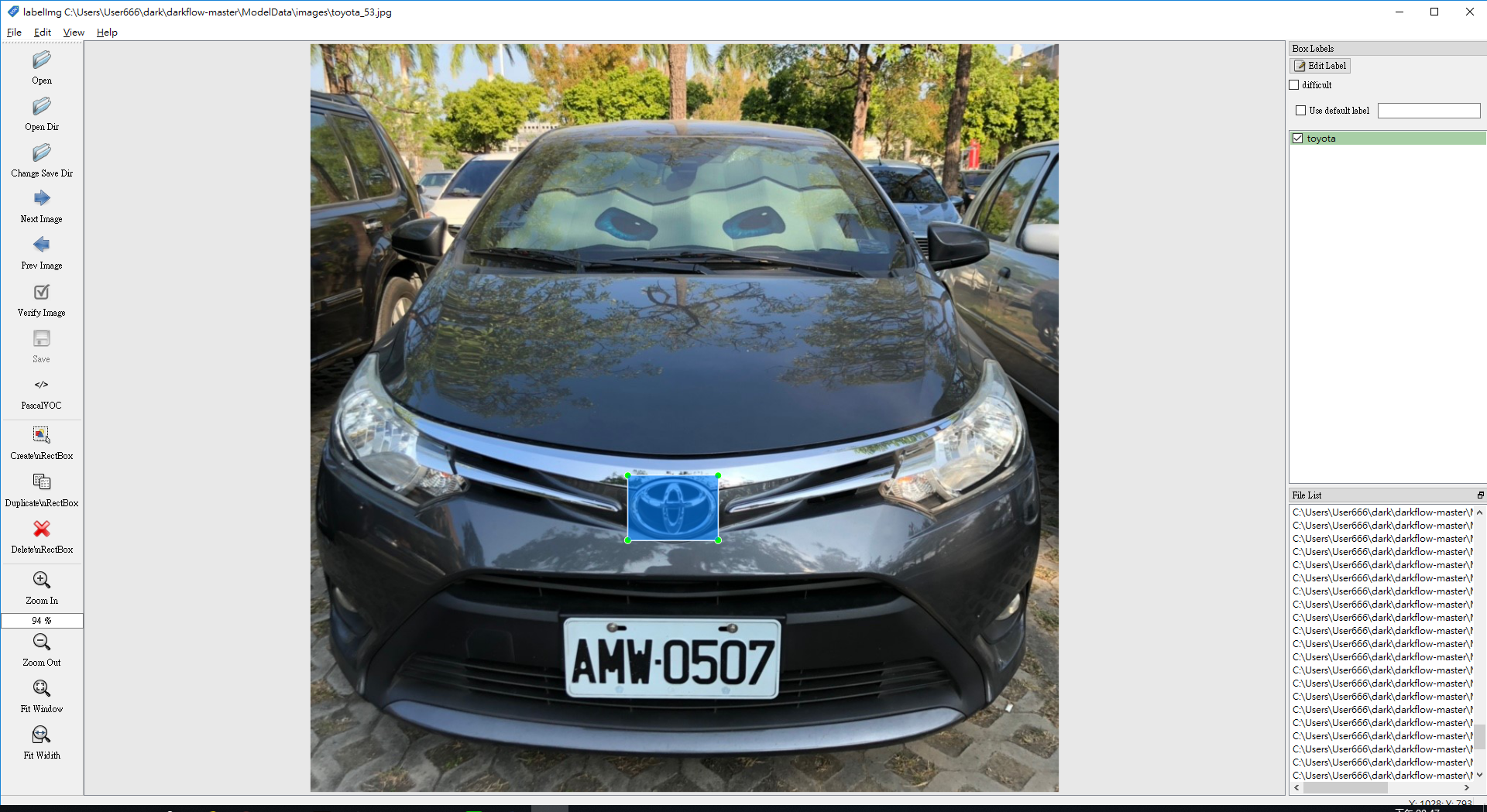
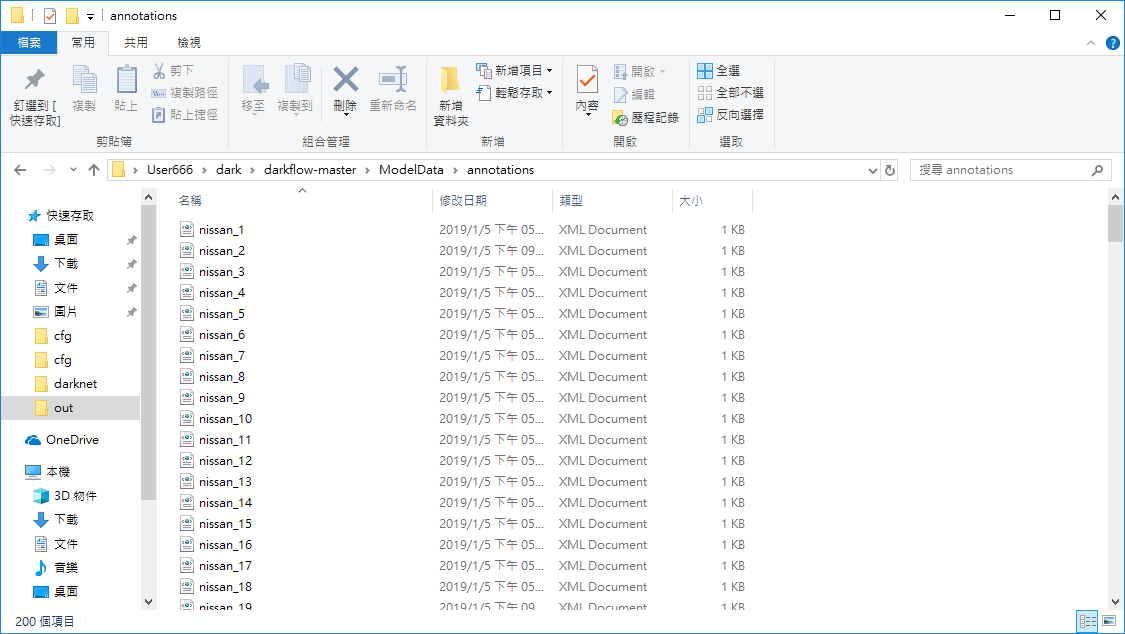


圖8、標記選取要辨識的部位

圖9、標記完會在事先設定好的儲存路徑出現XML檔

第三步：使用Yolov2-tiny模型開始訓練(如圖10、11、12)

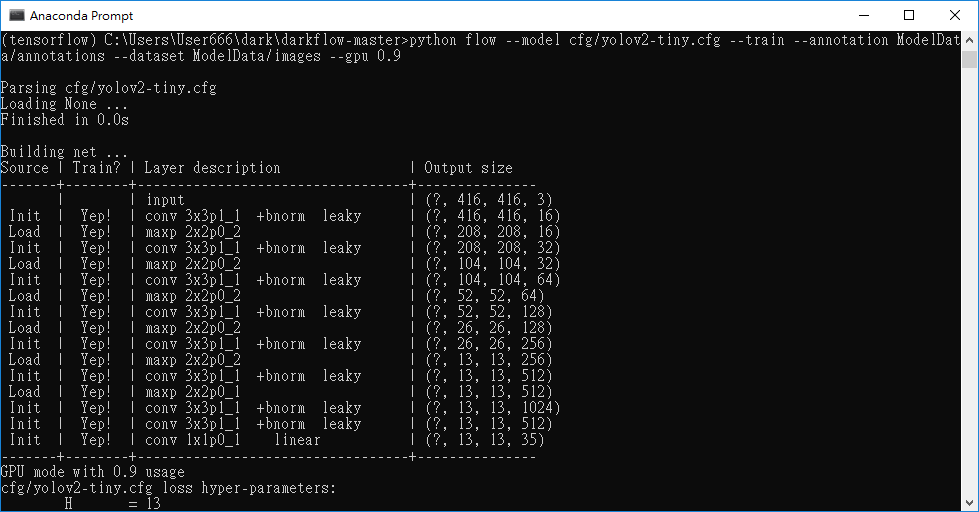


圖10、開始訓練

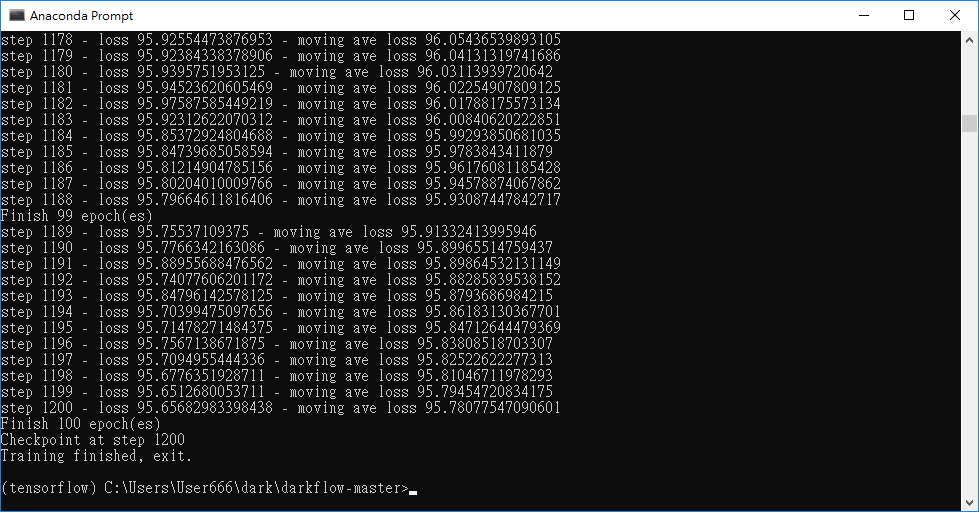


圖11、100次訓練完成

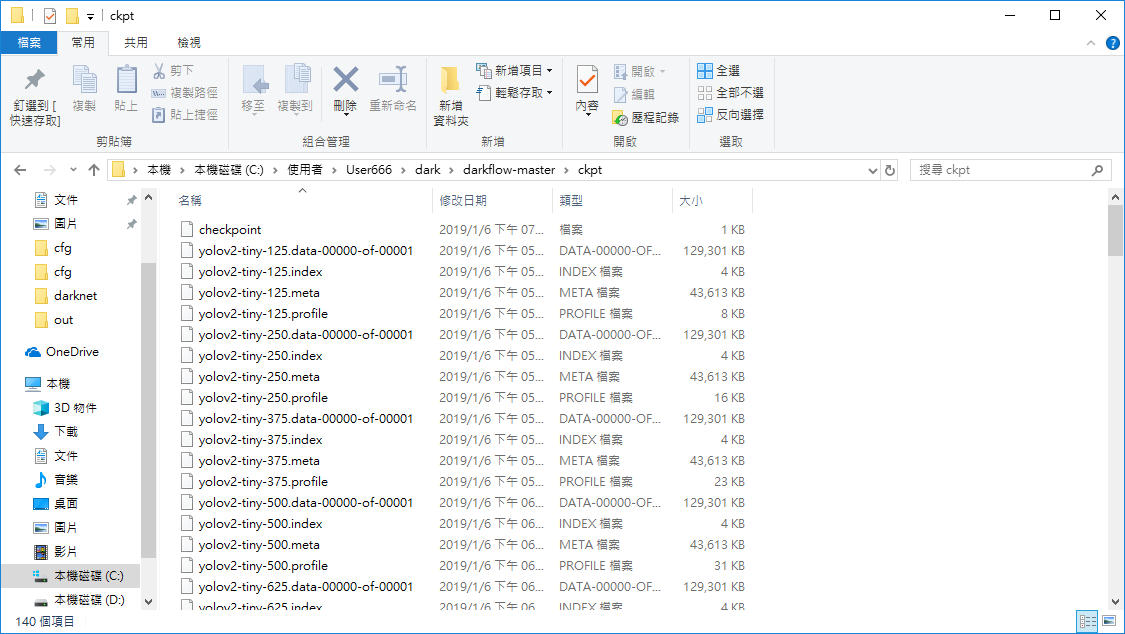


圖12、訓練完的資料儲存到checkpoint

第四步：使用訓練好的資料開始辨識圖片(如圖13、14、15)



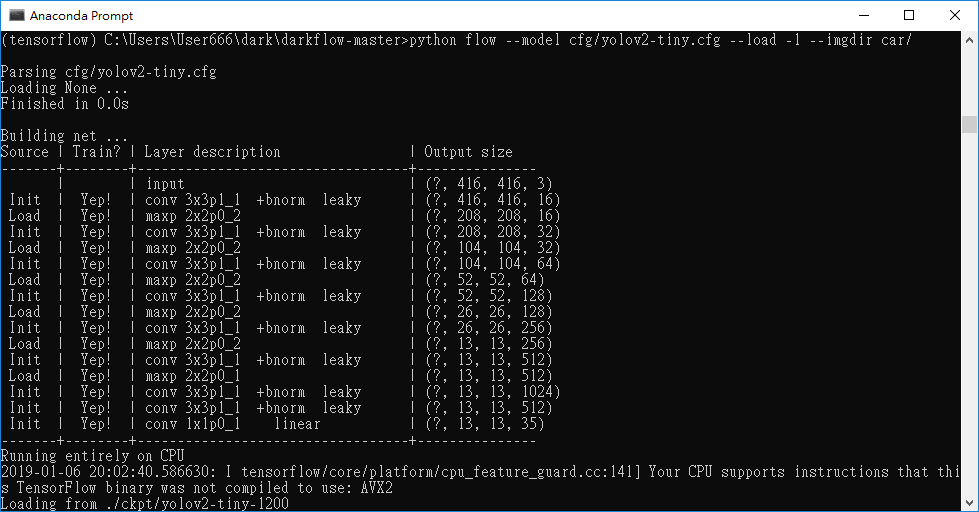
圖13、testing的資料一類20張

圖14、執行測試

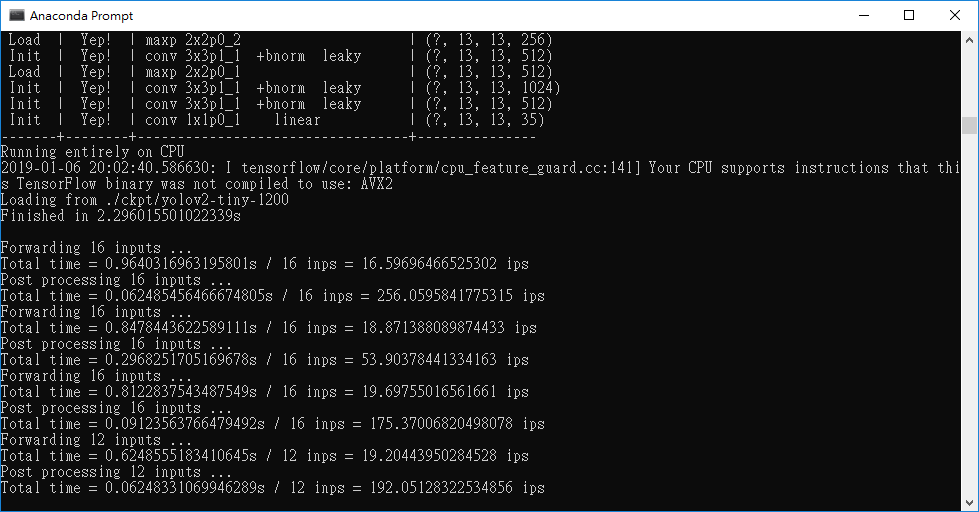


圖15、測試完成

第五步：辨識結果輸出(如圖16、17)

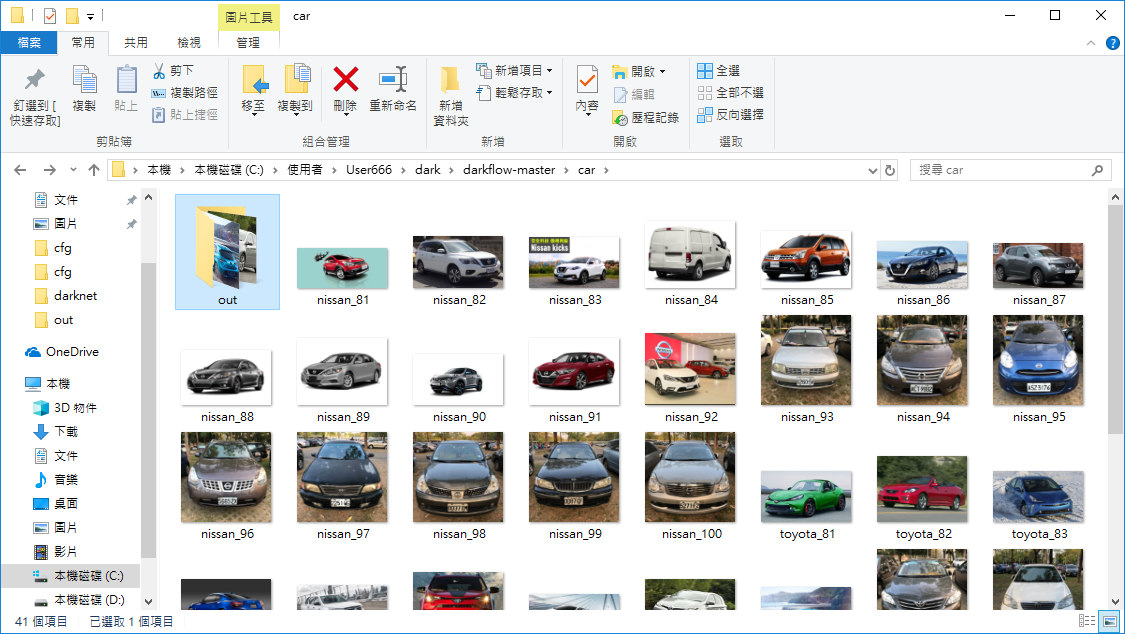


圖16、出現Out輸出資料夾

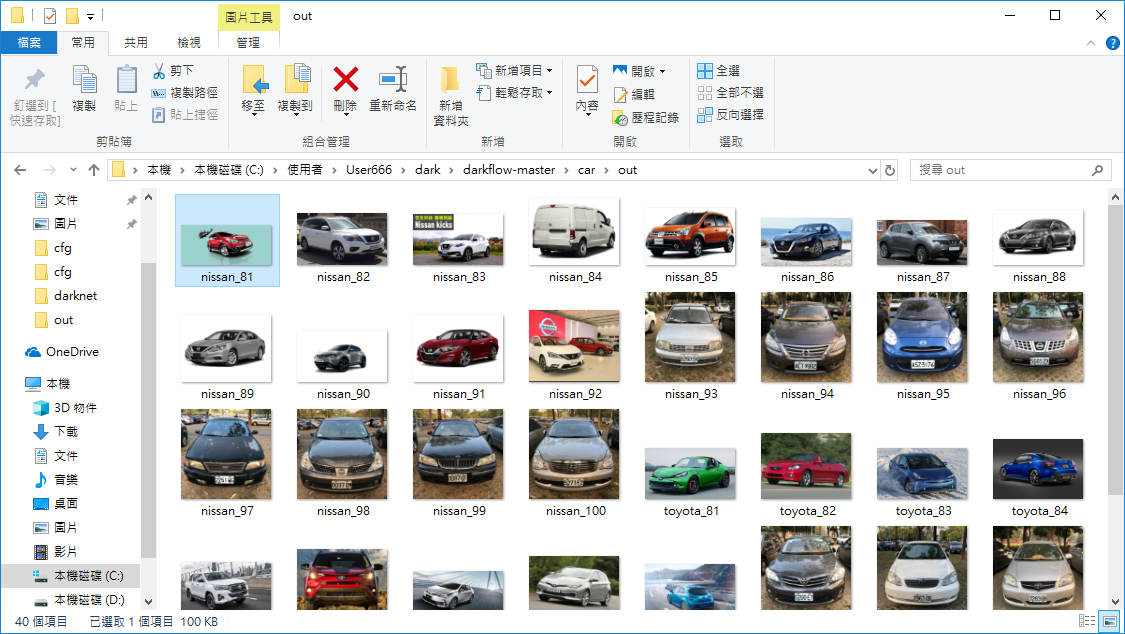


圖17、辨識結果

1. 實驗結果與討論

**(一)資料集介紹**

我們在學校拍攝了200張的汽車車頭影像並從中篩選出特定廠牌，各類不足的樣本數目，則是從網上搜尋，資料集內總共為200張影像。我們將影像分類成兩類，第一類是TOYOTA廠牌的汽車，第二類則是NISSAN廠牌的汽車。

**(二)實驗結果**

表1、訓練過程詳細一覽

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 各類訓練張數 | 各類測試張數 | 訓練次數 | 影像解析度 |
| 第一次訓練 | 70張 | 30張 | 650次 | 原圖 |
| 第二次訓練 | 70張 | 30張 | 100次 | 300x300 |
| 第三次訓練 | 70張 | 30張 | 100次 | 原圖 |
| 第四次訓練 | 80張 | 20張 | 150次 | 原圖 |
| 第五次訓練 | 80張 | 20張 | 150次 | 原圖 |
| 第六次訓練 | 100張 | 20張 | 100次 | 原圖 |

1. 結論

在本次實驗中，我們嘗試使用YOLOv2來辨識出汽車logo，經過多次修改及嘗試，仍舊無法成功辨識出來，我們推測是資料集的數量不夠、訓練次數過少、影像拍攝角度不夠全面，又或者是我們訓練資料集的方向錯誤，當然程式能力不足也是其原因之一。

而在windows作業系統上，使用darknet的資源，要預先準備的資源冗長，可能有那裡漏掉、安裝不全、環境變數設定錯誤等等的各種原因，導致程式出錯，而這些點在網路上能找的資料不齊全，礙於能力不足，很難去做修正。

1. 參考文獻

[1]Ebrahim Karami, Siva Prasad, and Mohamed Shehata, Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images

[2] Pavol Bielik, Veselin Raychev, Martin Vechev, PHOG: Probabilistic Model for Code

[3] Haiyuan WU, Yukio YOSHIDA and Tadayoshi SHIOYAMA, Optimal Gabor Filters for High Speed Face Identification

[4] Mustafa S. Kadhm, Lim Shu Yun, Propose A Simple and Practical Vehicle Logo Detection and Extraction Framework May – June 2015 pp87-90

[5] M. Naveenkumar, A. Vadivel OpenCV for Computer Vision Applications, Proceedings of National Conference on Big Data and Cloud Computing (NCBDC’15), March 20, 2015

[6] Su-wen ZHANG, Yong-hui ZHANG, Jie YANG, Song-bin, Vehicle-logo Recognition Based on Convolutional, 2016 International Conference on Computer, Mechatronics and Electronic Engineering (CMEE 2016)

[7] Malisa Huzaifa,Iping Supriana Suwardi, Car Model Recognition from Frontal Image Using BRISK, International Conference on Electrical Engineering and Computer Science (ICECOS) 2017 pp104-106

[8] Stefan Leutenegger, Margarita Chli and Roland Y, BRISK: Binary Robust Invariant Scalable Keypoints

[9] Martin A. Fischler and Robert C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography

[10] Deepak Geetha Viswanathan, Features From Accelerated Segment Test(FAST)

[11] Hiroki Nakahara, Tomoya Fujii, Haruyoshi Yonekawa, Shimpei Sato, A Lightweight YOLOv2: A Binarized FPGA 2018, February 25–27, 2018, Monterey, CA, USA

CNN with A Parallel Support Vector Regression for an FPGA

YOLOv2的原理與實現 https://zhuanlan.zhihu.com/p/35325884

Darknet-19 https://pjreddie.com/darknet/imagenet/