Final Project - Detect Vehicle Types

B06705026 林語萱
B06705008 戴聆

前言

由於科技的發展與進步，自駕車的技術越來越成熟，本組便以此做為發想，希望可以用到課堂上學習的物體偵測技術，結合未來自駕車的發展，作為本組期末專題的方向。由於本組希望專題成果可以真實應用到日常生活之中，本組最終決定以台灣道路上的交通工具為主題去建立使用深度學習技術的物體偵測模型。期望未來可以將本組的成果延伸到車流控制、車道分流、道路規劃及管理、緊急車輛路線規劃、交通號誌控制、科技執法等不同面向的應用。本組最終選擇 SSD(Single Shot MultiBox Detector)作為此次車種辨識的訓練模型。

文獻回顧

Reference paper : Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision 2016.

SSD 採用 VGG16 作為基礎模型。然而，當網路層數越深，小物件在經過多次的捲基層迭代後能留下來的訊息就越少，導致最後對於小物體的預測效果不好。因此 SSD 模型再加上一些卷積層來加深網路、在不同大小的特徵圖進行預測，並且為每層特徵圖設置不同大小和長寬比的 anchor 以預測不同大小的目標。這些卷積層的大小是逐層遞減的，在較低層設置較小的 anchor，能夠提取出更精細的特徵。

在預測時，對於每個 anchor 會預測生成框的偏移量(cx,cy,w,h)及每個類別的分數，並調整 anchor 的偏移量以更好的匹配物體形狀。而 SSD 的損失函數分為兩類，分別用 cross entropy 作為分類的損失函數，並用 smoothL1Loss 來作為 bounding box 的 regression。

最後要進行輸出時，會將不同捲積層上的 bounding boxes 進行 Non-maximum suppression 得到最終的預測結果，也就是一系列 bounding boxes、預測的物體種類與 score。而較低層的捲積層對小物件的偵測較好；越後面的捲基則較可以掌握大物件的訊息。

而本組之所以選擇 SSD 作為訓練模型，在於該篇論文中的實驗證實，SSD 的訓練過程相較於其他物體偵測模型更快速，並且偵測結果的準確率更高。SSD 訓練過程更快，因為 SSD 用 anchor 機制取代了 selective search，並將所有計算封裝在單個 CNN 網絡中，並且 SSD 模型沒有對 anchor 進行 re-sampling。而 SSD 在每個位置使用 3x3 等較小

的捲積核、對於不同長寬比的 anchor 使用不同的捲積核，可提取出精細的特徵，並且在多層 feature maps 上對不同大小的 bounding box 進行預測，因而能提升準確率。論文中提到的這些改進設計，也能夠在圖像解析度較低時，保證偵測的精確度。

資料來源

本組選擇 Imagenet 公開資料集作為本組的資料來源，不僅是因為它容易取得下載，資料類別也很豐富，與本組題目相關的交通工具類別多達十三個，而且每個類別中的圖片的角度與背景也很多元，可以提高本組模型訓練的準確率。

13 個偵測類別：

類別名稱	類別編號
Ambulance	n02701002
Compact car	n03079136
Fire truck	n03345487
Garbage truck	n03417042
Minibus	n03769881
Minivan	n03770679
Motor scooter	n03791053
Mountain bike	n03792782
Moving van	n03796401
Police car	n03141065
Taxi	n02930766
Tow truck	n04461696
Trailer truck	n04467665

本組先是寫了個 python 程式把 Imagenet 網站提供的 URL 圖檔連結爬下來，再把對應到的圖片下載存成 JPG 檔。透過人工檢查的方式，把不像台灣會出現的交通工具的圖片刪除，因為這樣可以讓本組的訓練資料集與真實狀況的資料更相近，測試應用時的準確率也會更高。最後本組篩出十三個類別共 3437 張圖片作為本組資料集的內容。

關於圖檔的前處理，本組將 3437 張圖片全部調整成 300x300 的大小，再按照 000000.jpg 到 003436.jpg 的編碼方式重新命名。最後用 labelImg 程式以人工的方式一張一張為圖片標上 bounding box 並且生成對應的 xml 檔。

上圖為 labelImg 程式的畫面

製作資料集

本組將所有的圖檔以 7 : 3 隨機分成 training data + validation data 與 testing data，再 training data + validation data 以 8 : 2 隨機分成 training data 與 validation data。並且生成所有類別記錄 training、validation 與 testing 的 txt 文件檔。本組是以 Pascal VOC 的格式製作本組的資料集的，也就是說，本組的資料集會是以以下的結構組成的：

- VOC2007
  - Annotations (XML bounding box files)
  - ImageSets/Main (txt files)
  - JPEGImages (JPG images files)

在 VOC2007 資料夾底下會有 Annotations、ImageSets、JPEGImages 資料夾，Annotations 底下要放置標示 bounding box 的 xml 檔，ImageSets 底下有一個 Main 資料夾底下放置每個類別的 Training、Validation、Testing 的 txt 檔，JPEGImages 底下放置 JPG 圖檔。這樣便完成本組資料集的製作了。

模型訓練

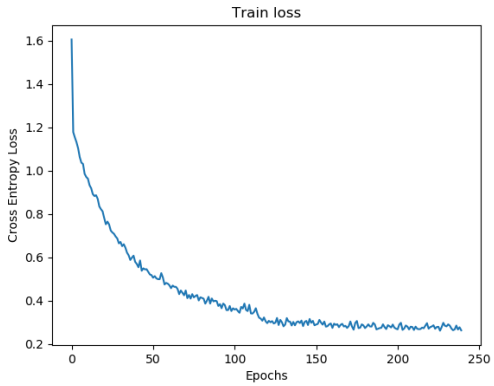
本組的車種辨識模型是基於 https://github.com/zhreshold/mxnet-ssd.git 所提供之 SSD 模型，再根據本組製作的資料集與其他所需條件對模型進行相對應的修改。
本組模型的訓練環境是使用 Google 為深度學習研究提供的 Google Colab 環境，訓練的程式碼是以 python 3 完成的，模型的訓練的框架是使用 MXNet，並且使用 Nvidia 的 CUDA 技術達成 GPU 的加速計算。

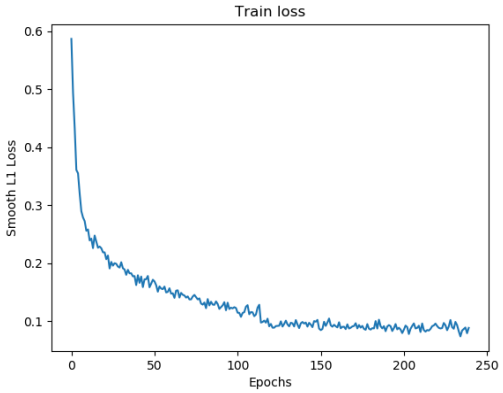
訓練過程的參數設定是 240 個訓練 epoch，batch size 設定為 32，optimizer 使用 sgd，weight initialize 是用 Xaivier，learning rate 一開始設定為 0.001，經過 120 epoch 之後 learning rate 降為 0.0002，模型的 training loss 更加地收斂。

訓練結果與討論

1. Training loss

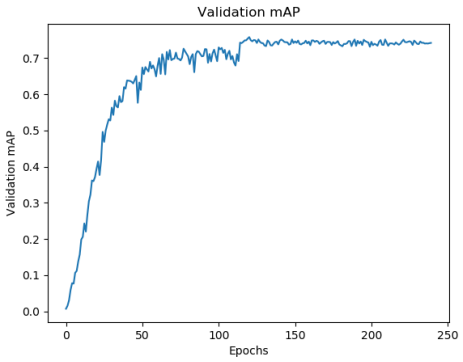
本組總共訓練了 240 個 epochs，最終模型達到收斂：cross entropy loss 收斂到 0.28 左右，smoothL1Loss 則收斂到 0.08 左右。





2. Testing

經過 240 個 epochs， validation mAP 收斂並趨於 74%左右。

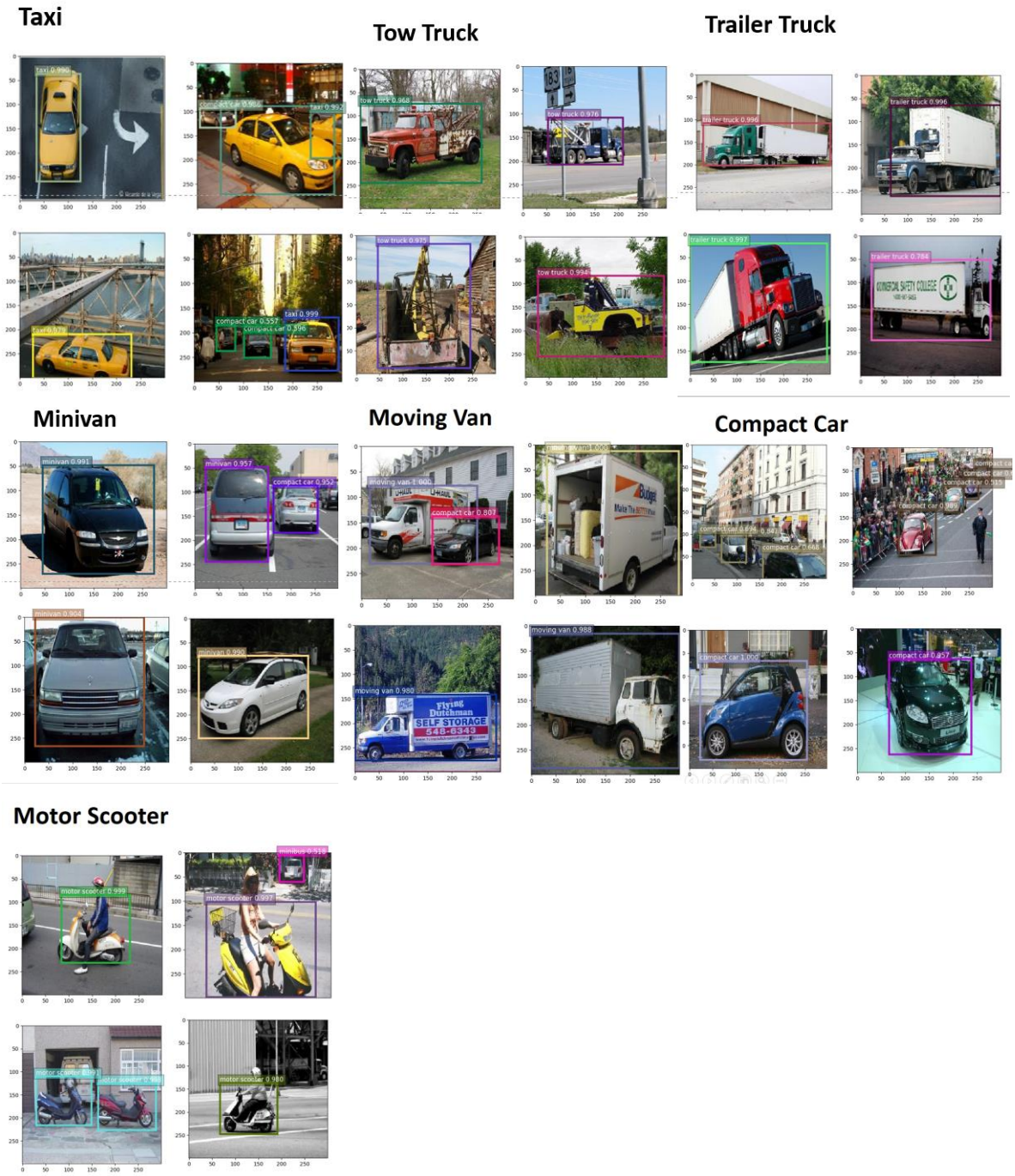


而從各個類別來看，最低的是 trailer truck 類別，只有 55%左右。最高的則是 firetruck 類別，達到 0.89%，而 mountain bike、taxi 以及 motor scooter 類別也達到 80%以上，推測是因為以上類別的車輛型態有比較明顯的特徵，因而與其他類別較不容易混淆。

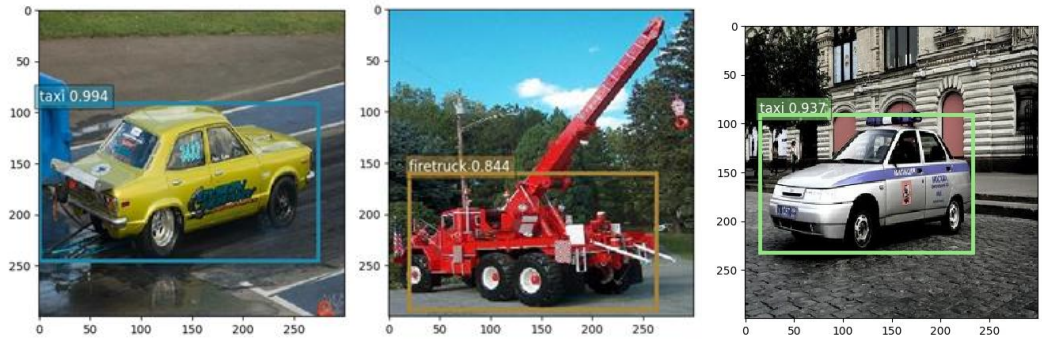
- firetruck = 0.892236
- mountain bike = 0.875299
- taxi = 0.848488
- motor scooter = 0.803890
- ambulance = 0.798887
- garbage truck = 0.767433
- police car = 0.782929
- minibus = 0.750458
- tow truck = 0.715817
- minivan = 0.661453
- moving van = 0.647940
- compact car = 0.587256
- trailer truck = 0.551899

3. Testing - Images

本組從 testing dataset 中每個類別選取圖片，調整成 300x300 的大小並餵進本組訓練好的模型，以測試並視覺化模型的類別預測、bounding boxes 預測與 score 準確度結果。模型成功預測物體類別的準確率大約在 7 到 8 成，此外觀察發現，當物體較小時 bounding boxes 雖然也能夠框住物體，但預測稍不準確。以下展示一小部分測試結果。



而物體類別預測錯誤部分，本組觀察多半是因為該物體的顏色或車種與其他類別的多數車輛型態相似而被誤判，例如下圖中 compact car 被誤判為 taxi、tow truck 被誤判為 firetruck、police car 被誤認為 taxi 等等。



4. Testing - Video

本組錄製了一段約 17 秒的影像，並用 cv2 函式庫對影像進行處理。讀入影像後將影像每隔 2 幀儲存下來，並將這些影格調整成 300x300 的大小。接著將影格餵進本組訓練好的模型，若預測出物體的 score 大於 0.5，則畫上預測的物體類別、score、bounding boxes。最終再將這一幀幀影格輸出成影像呈現。

影片 demo:
https://drive.google.com/file/d/1EBlQkFpeDIaobJshDwvHeQn6w_NbCfzH/view?usp=sharing

影片中可以成功預測到物體與類別，並將類別、bounding boxes 與 score 呈現在畫面上。不過觀察發現，當物體較小時 bounding boxes 的預測稍不準確，例如圖 1 中 Minubus 類別雖有預測正確，但 bounding boxes 並沒有框住整輛公車。此外，隨著物體較所占螢幕越大，物體預測的 score 也會逐漸上升，例如圖 2 到圖 4，minibus 的 score 從 0.57 上升 0.92 再到 0.95，並且 bounding box 越加準確，而 motor scooter 的 score 也從圖 2 的 0.74 上升到圖 3 的 1.00。

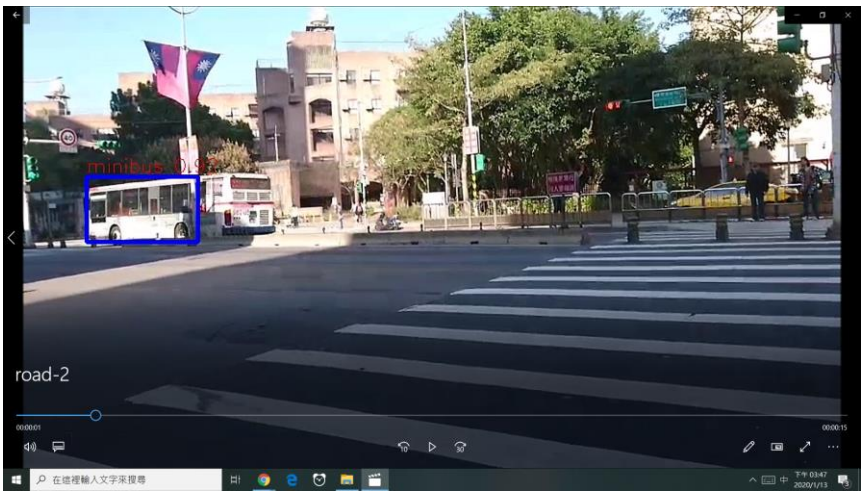


圖 1

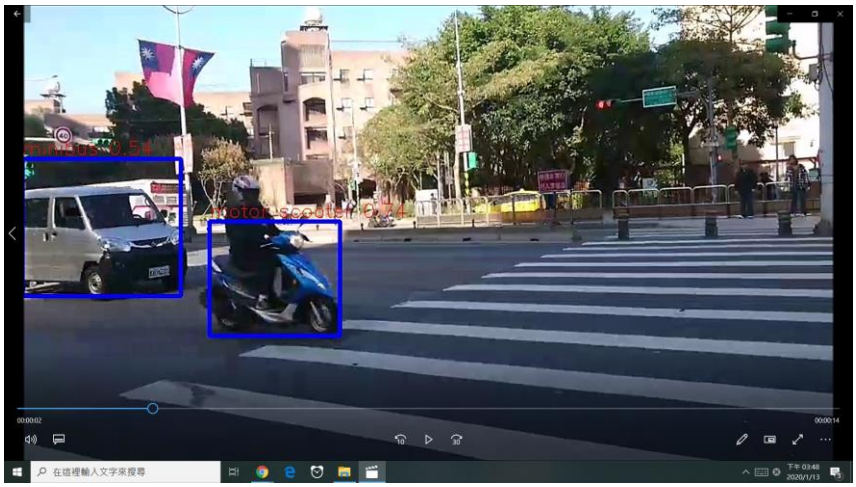


圖 2

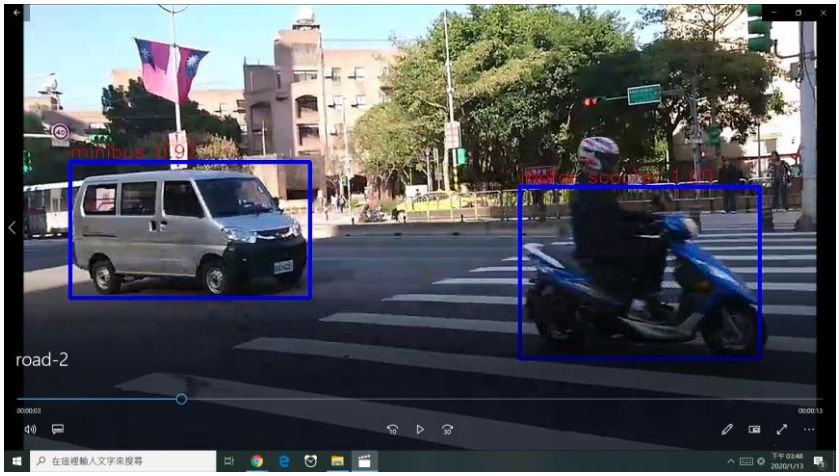


圖 3

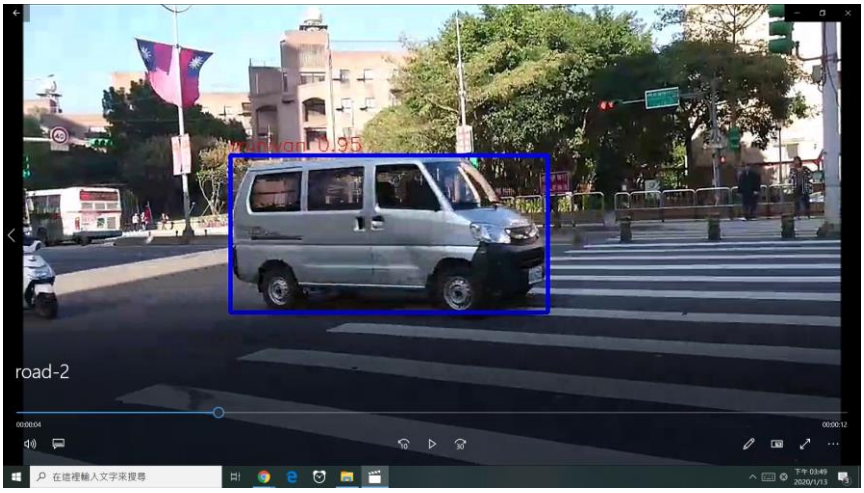


圖 4

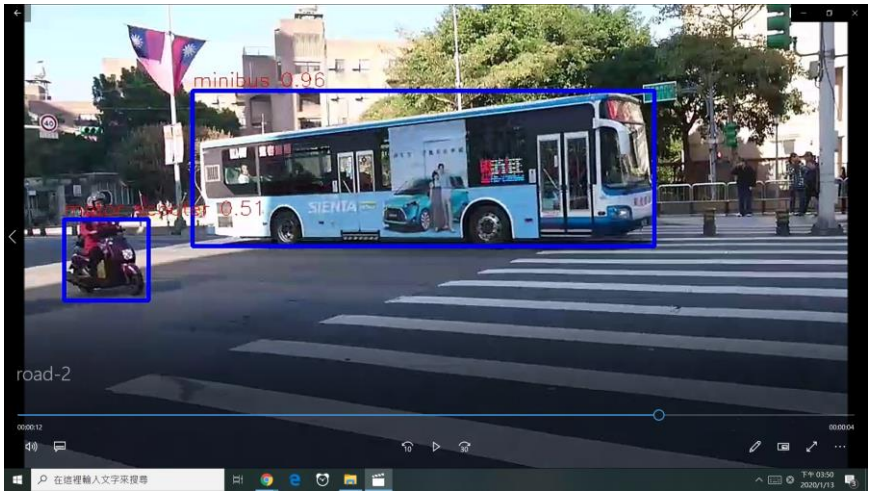


圖 5

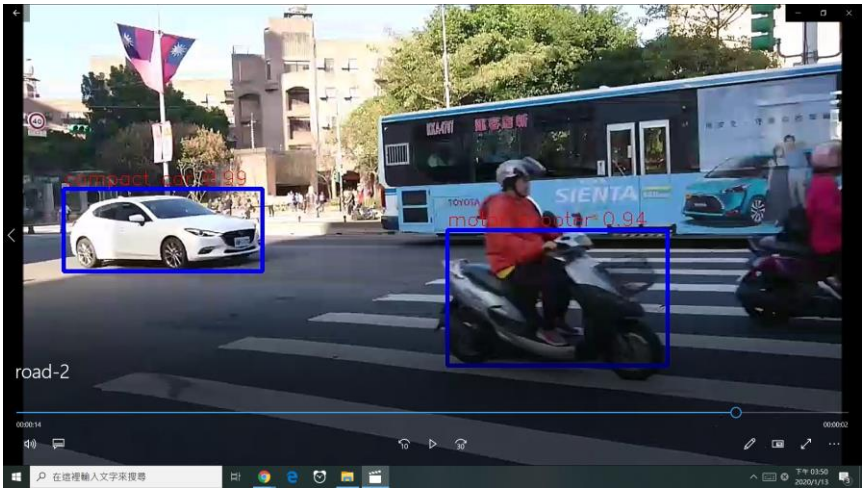


圖 6

結語

模型的訓練資料集來源為 IMAGENET，而 IMAGENET 提供的圖片相當多元，所以模型訓練結果不錯，透過圖片與影片證實能夠實際偵測出車種。

此外，為了建構更加準確的物件偵測模型，未來可能可以透過建構更強大的訓練資料集、修改網路架構等方式再降低 cross entropy loss、smoothL1Loss，並提升 mAP。

回顧本次 project 動機，本組希望建構出來的模型能夠實際應用，其中也包括了實現實時的物體偵測。然而讀入影格、進一步處理加上模型運算會花上一些時間，模型偵測物體的運算速度相對來說還是不夠快，目前似乎還難以實現實時物體偵測的目標。