**Deep-Learning-for-Autonomous-Driving**

Lab3 : Object Detection

ID : 310605007 Member : 鄭晴立 Department : 機器人學程

# Introduction

首先，本次的時間較為緊迫，而目前Yolov4在研究上已經有不錯的準確度及速度表現，因此我評估後認為，對於各種參數的修正或是model的改進皆難以達成顯著的效果。最終我採用Yolov7去嘗試實作。

# Yolov4

## Implement

由於沒辦法一一嘗試各項參數影響，因此在Yolov4的實作中，基本上就是講文獻中提到的方法都有使用，例如：mosaic為Yolo中很重要的data augmentation；pretrained model及Cosine learning可以大幅提升training的穩定度及速度。

表 一Yolov4參數設定

|  |  |
| --- | --- |
| Parameter | |
| Backbone | Mobilenetv3 |
| Optimizer | Adam |
| Learning rate | StepLR |
| Epoch | 300 |
| Loss function | CIoU |
| Batch size | 16 |
| Smooth label | 0 |
| mosaic | True |
| Cosine learning | True |
| pretrained | True |
| Unfreeze Epoch | 50 |

## Conclusion

如我所預估的Yolov4已經有不錯的效果，即便我嘗試不同的model及參數，皆無法有效的提升，而在時間有限的情況下，我決定直接常是yolov7，因此下一章節介紹，我使用yolov7的結果。

|  |  |
| --- | --- |
| 一張含有 文字, 路, 路面, 景色 的圖片  自動產生的描述 | 一張含有 景色, 路, 路面, 室外 的圖片  自動產生的描述 |
| 一張含有 文字 的圖片  自動產生的描述 | 一張含有 文字 的圖片  自動產生的描述 |

圖 一 Yolov4偵測結果，下方為其表現數據和方框座標

# Yolov7

由於上述原因我嘗試使用Yolov7，而由於Yolov7是Pytorch架構，因此理論上可以直接用上，但後來在Datasets部分遇到問題所以無法實現。

## Datasets

然而，yolov7在data處理上有極大不同。在Yolov7中所使用的coco datasets的4個座標分別是長、寬跟物體中心的座標，而助教所給的highway datasets則是bounding box左上角的座標及右下角的座標，如下圖。。

除label方式的差異外外，Yolov7的匯入資料方式也有極大不同，可以看到yolov7是一個圖片對應一個txt檔，每一個txt檔裡面則記錄一個圖片的label，如下圖。

|  |  |
| --- | --- |
|  |  |

圖 二 yolov7 data

一張含有 個人, 牆, 室內, 女性 的圖片

自動產生的描述

圖 三 以這張圖為例，在Yolov7 label的座標格式下，左邊的人為(0.31, 0.4, 0.62, 0.97) ;領帶為 (0.35, 0.57, 0.09, 0.40)，以左上為原點，表示為x、y、寬、高，對應於整體邊界的比例，可以看出其中心點為物體正中央

一張含有 文字, 路, 路面, 景色 的圖片

自動產生的描述

圖 四 以這張圖為例，其為416x416，在Yolov4 label的座標格式下，右邊的Auto為(222, 140, 284, 287) 、下面的Auto為 (118, 182, 215, 412) 、上面的Car為 (206, 115, 231, 147)、下面的Car為(199, 129, 226, 173)，以左上為原點，表示左上點的座標及右下點的座標。

## Implement

由於Datasets的差異，沒有足夠時間調整Data的labels，因此只好作罷，直接使用Coco datasets train出的Yolov7去使用，可以看到，我使用100張圖片去偵測，僅花了2.972秒。

而實際辨識成果上，由於是用Coco datasets train出來的，可以看到會有很多不同的東西被偵測出來，以下面四張圖為例，他會很容易辨識到路人，較大的車輛比較容易辨識成功，而深色的車輛容易辨識乘火車，聯結車則會偵測成公車。

而從confusion matrix可以看出80樣coco的分類都有很好的結果，而在上一次lab的data中也蠻不錯的，在日常生活的照片也有不錯的成果。



圖 五 yolov7偵測100張圖片所花的時間

|  |  |
| --- | --- |
| 一張含有 景色, 路, 路面, 室外 的圖片  自動產生的描述 | 一張含有 路, 景色, 路面, 公路 的圖片  自動產生的描述 |
| 一張含有 文字, 路, 路面, 景色 的圖片  自動產生的描述 | 一張含有 路, 景色, 路面, 室外 的圖片  自動產生的描述 |

圖 六 yolov7偵測成果

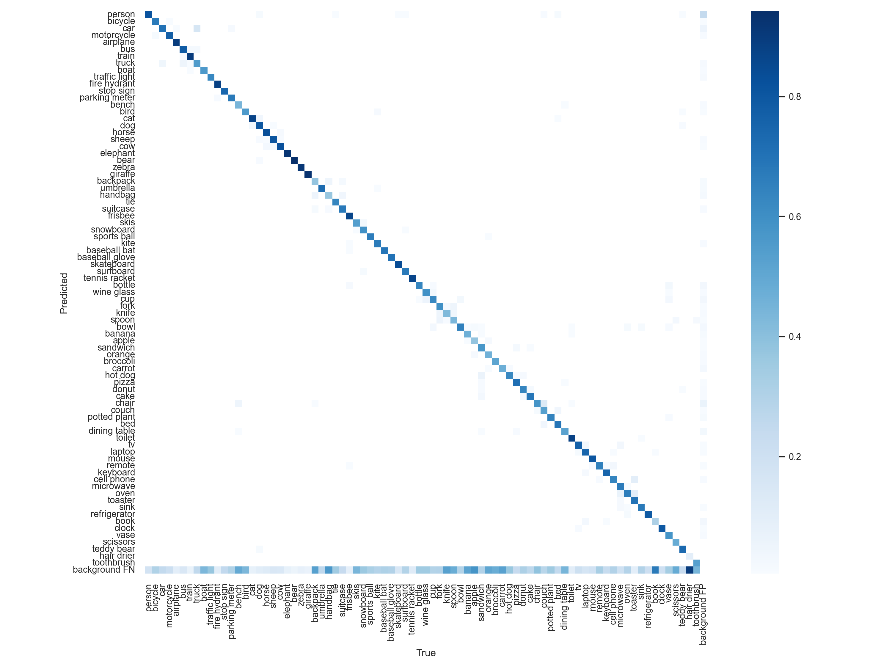


圖 七 yolov7的confusion matrix

|  |  |
| --- | --- |
| 一張含有 室外, 建築物, 公車, 路面 的圖片  自動產生的描述 |  |
| 一張含有 文字 的圖片  自動產生的描述 | 一張含有 文字, 路面, 室外, 天空 的圖片  自動產生的描述 |

圖 八 Yolov7的其他偵測成果，左下為我房間，右下為上一次lab的data

# Discussion

## 原先的結果已經不錯，難以有顯著提升

Yolov4的精準度已經有不錯的成果，而本次實驗的data也較為單純，因此不做任何修正就可以有不錯的表現，改變backbone或是任何參數修正，皆無法達到顯著的效果，甚至會拖慢其辨識速度，而yolo本身的特色在於其便是速度，因此，我認為太著重於精確的調整是沒有意義的，所以我才會直接以yolov7作為參考去研究。