

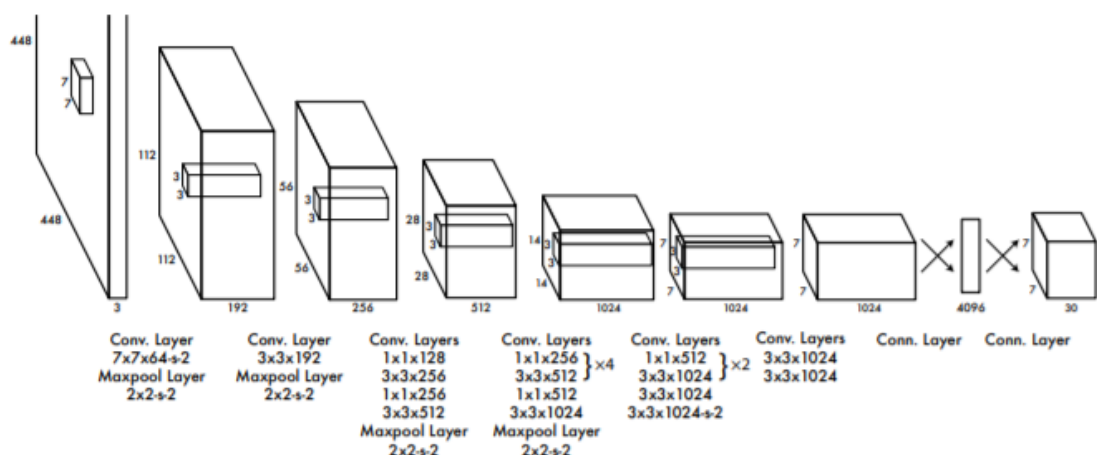
AI 之 YOLO 研究報告

前言一

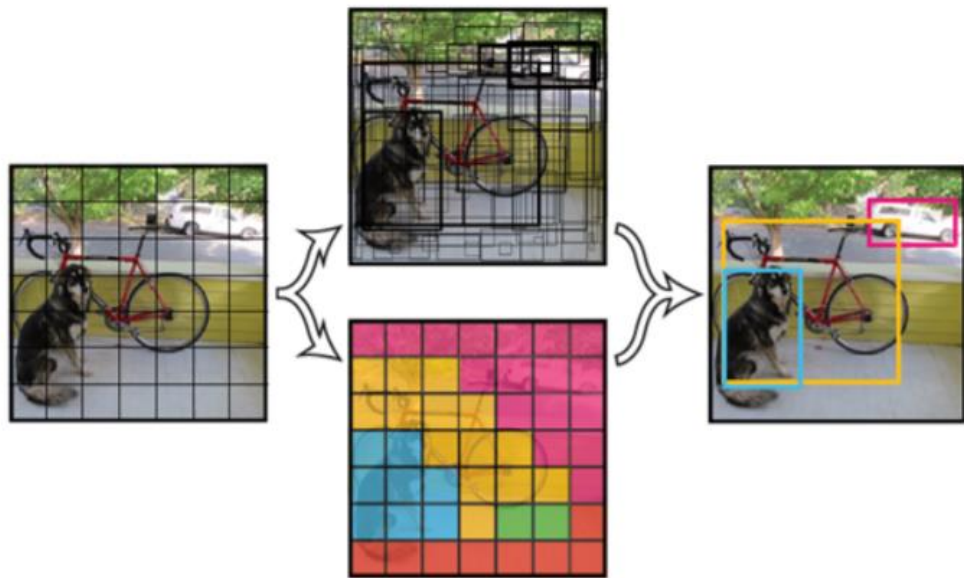
這是在研究 YOLO: Real-Time Object Detection 時，寫的研究報告。本研究以 YOLO 之操作方式和其原理作探討與討論。

內文一

YOLO 技術是利用先前所訓練好的資料庫，在遇到相片或影片時能夠在畫框中辨識特定物件的方法，由於以往追蹤物體的方法，大多採用比較畫面特徵點並搭配前後圖比對來分辨出前景與背景的差異，雖能找出能動的前景但無法確認該前景的分類，且因為要比對前後圖的關係導致整體追蹤速度很難達到實時運(30fps)。本研究所使用的 YOLO 版本為 YOLOv3，也就是第三代的 YOLO 系統，<引用>而在最一開始 YOLOv1 便可以辨識 20 種物體，其想法是將整張圖片作為輸入，直接預測出物體的位置與分類。方法是先將輸入圖片統一縮放成 448X448 的大小，再對圖片做卷積神經網路(Convolutional Neural Network，CNN)，如圖一經過 24 個卷積層與 2 個全連接層後，輸入圖片會變成一個 7X7X30 的矩陣，可以想像成輸入圖片會區分成 7X7 的區域，而 30 這個數值的由來是因為每個區域會預測 2 個邊界框(Bounding box)，每個邊界框要預測參數 x、y、u、h 與 confidence 共五項，這五個參數分別表示:(x, y)這兩者是物體外框的中心點;(u, h)這兩者是物體外框的大小;信心值(confidence)，代表與分類符合的程度。且每個區域會有 20 個類別預要預測，也就是(2X5+20)共 30 個參數，最終會輸出成 7X7X30 的 Feature map。每個區域會計算這兩個邊界框的中分類的信心值，找出最符合這個區域的分類，如圖二。</引用>



圖一、YOLOv1 的網路架構[1]



圖二、YOLOv1 的預測圖例[1]

CNN 原理

在網路上學習 YOLO 的文獻和原理時，時常看到 CNN(當然不是指新聞台)，現在就來剖析他是甚麼吧!

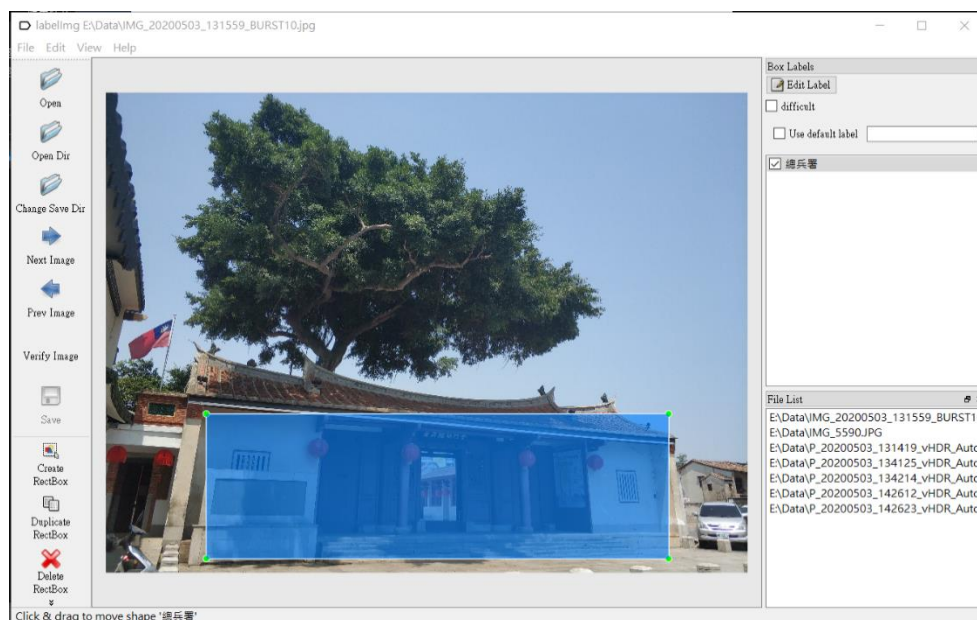
<引用>CNN (Convolutional Neural Network)是一種運用在圖像辨識/分類時，非常有效的網路結構。CNN 最主要的特點，就是它的卷積層(Convolution Layer)。我們在使用傳統演算法時，也常會使用到 Convolutional 方法，應用在影像的去雜訊、銳化、模糊化、抓邊緣等處理，它的運算概念，基本上和 Convolution Layer 是相同的。Convolution Layer 就是一種 2D 的 filter 運算，和傳統演算法的差別，在於它的核(Kernel)是在訓練中產生的，而設計傳統演算法時，這個核是我們給定、在程式碼寫死的。也就是說，我們在訓練 Convolutional Layer 時，其實就是在找出一個[對於訓練樣本，可以最有效提取特徵的 Kernel]。**[2] </引用>**

操作方式<自我實作流程，方法則是引用後歸納>

先以訓練方式做解說，因為是 AI 的辨識因此會需要用到 Pre-trained 的資料，大致上的運作都是照著以下網站執行實做: [YOLO3 darknet 训练自己的数据](#)。

步驟一：

在一開始，必須將拍下來需要訓練的圖片標上 Label，在這裡我們使用到 LabelImg，如圖三，將所有圖片都標上後便可繼續。



圖三、LabelImg 實作

步驟二:

將 xml 的標籤檔改成 txt 檔，在教學網站中也有附上 xmltotxt.py 的檔案可使用。

步驟三:

當你照著網站逐步完成後，會來到需要準備 yolov3-KD.cfg_train 檔案的狀況，這時候會發生一種問題，就是你訓練到一半會跳出，Out of memory，也代表著你的電腦負荷不了一次這麼多資料的進出運算，這時記得根據網站上更改 yolov3-KD.cfg_train 內的內容，將一次運作的數量減少，具體而言長這樣，每次計算的圖片數量 = batch/subdivisions。

步驟四:

當準備完成後，便可輸入: <不算是引用，因為它就只是指令碼>

```
darknet.exe detector train data/KD.data cfg/yolov3-KD.cfg_train darknet53.conv.74
```

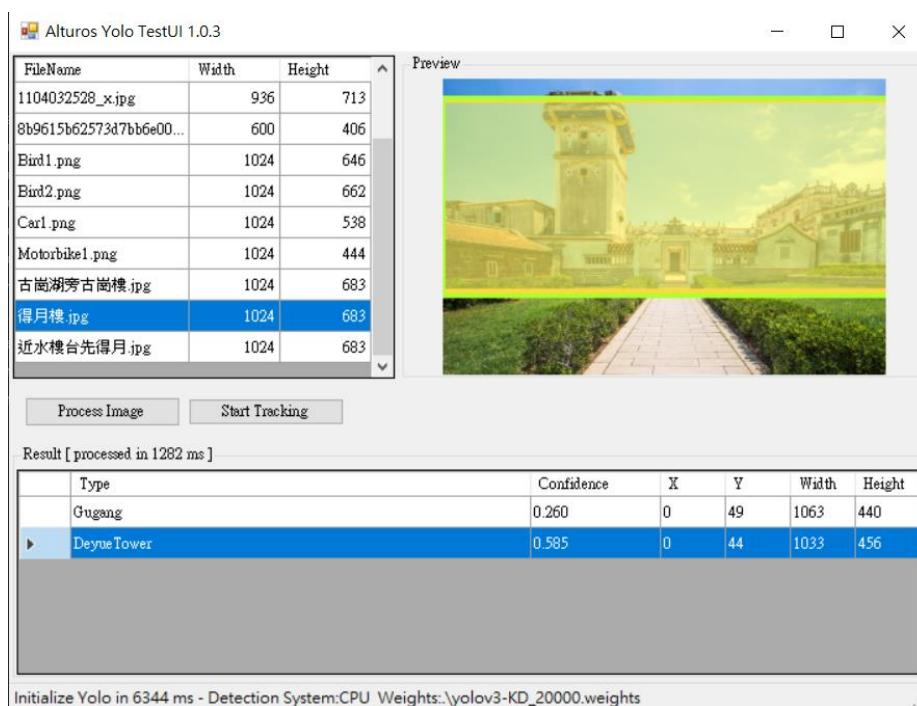
進行訓練，這麼一來就完成了。

Alturos.Yolo 套件

A state of the art real-time object detection system for C# (Visual Studio). This project has CPU and GPU support, with GPU the detection works much faster. The primary goal of this project is an easy use of yolo, this package is available on nuget and you must only install two packages to start detection. In the background we are use the Windows Yolo version of AlexeyAB/darknet. Send an image path or the byte array to yolo and receive the position of the detected objects. Our project is meant to return the object-type and -position as processable data. This library supports YoloV3 and YoloV2 Pre-Trained Datasets [3]

利用 C # 即時對象檢測系統 (Visual Studio) · 具有 CPU 和 GPU 支持 · 可在 nuget 上安裝及使用 · 路徑或字節數組發送到 yolo 並接收檢測到的對象的位置。將偵測到的類型和位置返回。將訓練後的 Weight、cfg 和 names 檔案放到 Alturos 內的:
E:\Aulturos\Alturos.Yolo-master\Alturos.Yolo-master\src\Alturos.Yolo.TestUI\bin\Debug

開啟 Alturos.Yolo.TestUI.exe



實作金門得月樓。

參考文獻——

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi."You Only Look Once:Unified, Real-Time Object Detection"
- [2] 深度學習影像分析原理說明
<https://tw.openrobot.org/article/index?sn=11181>
- [3] Darknet
<https://pjreddie.com/darknet/yolo/>