YOLOv4 Tutorial #4-

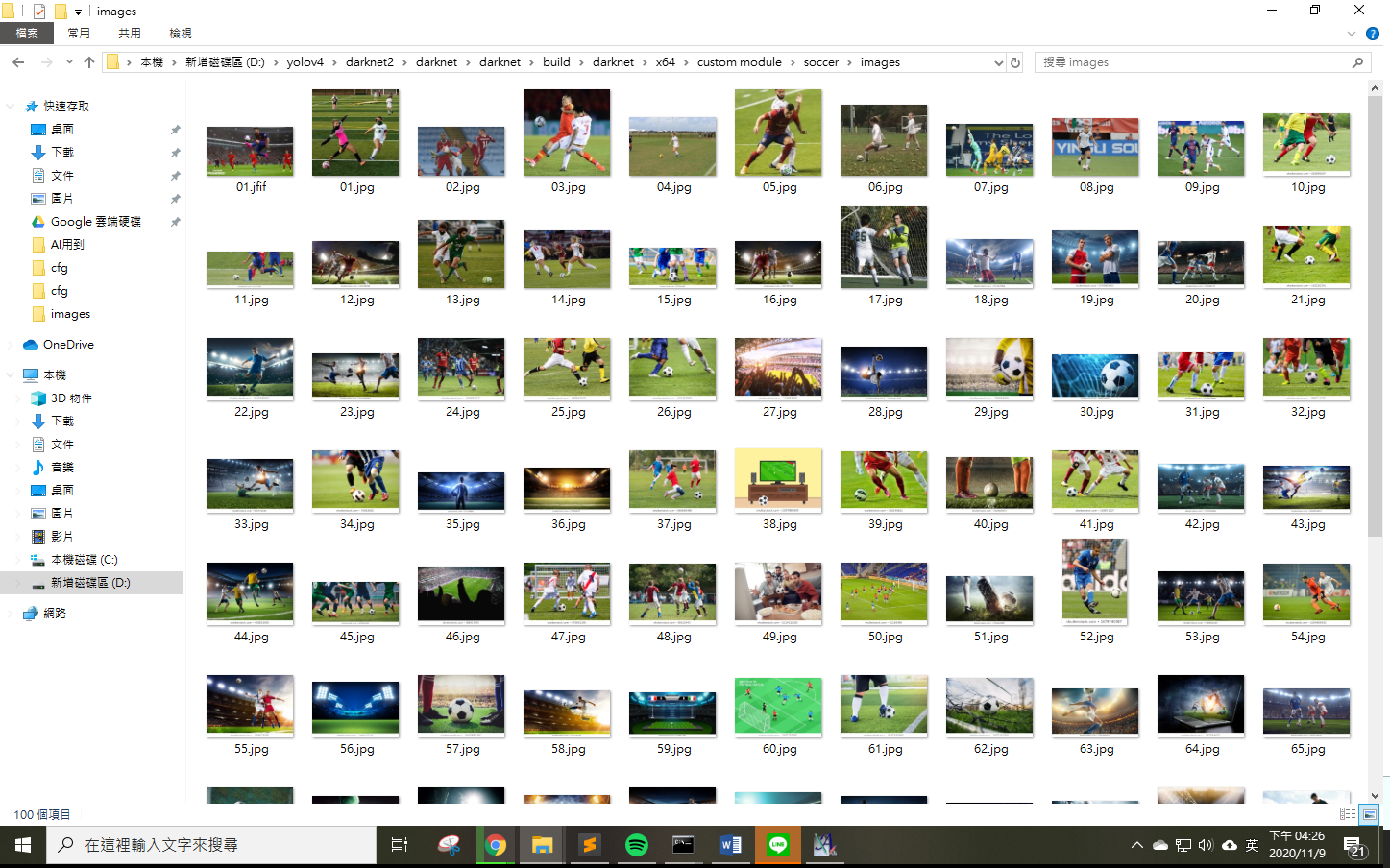
Traing Moudle

**前言：**

前篇已可利用v3的weight和cfg檔辨識基礎圖形，例如：人.手機等，而這篇要開始訓練自己得moudle，首先須取得圖形(影片內截圖或圖庫)，之後下載labelimg取得標記物件label的xml檔，透過convert2yolo程式將xml轉換為txt給yolo讀取，再輸入traing指令即可開始訓練

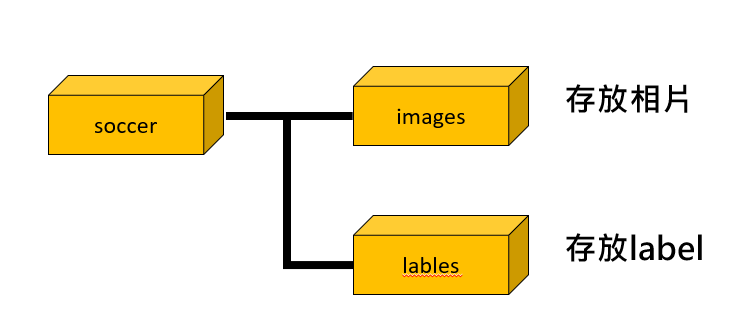
**圖片下載：**

前陣子再AI圈有個非常有名的大錯誤，足球比賽官方的AI錄影機應該追蹤足球，但由於裁判為光頭導致有很大部分攝影機都再拍裁判的光頭，這邊去GOOGLE尋找足球賽相關圖片，並透過[抓圖網站](https://imagecyborg.com/)與[RENAMER](https://www.azotaiwan.com/_dl_rC5CFd3nVq/_Mediafire_T0jpw7O3xD/AdvancedRenamerPortable_3.87_azo.exe_mediafire.htm)來加速找圖改檔名



**建立資料夾：**

在你的darknet x64內建建一資料夾專門來做此次實驗，並在下方創建兩資料夾，「images」專門放置相片，「labels」用來放置labelimg生成的xml檔

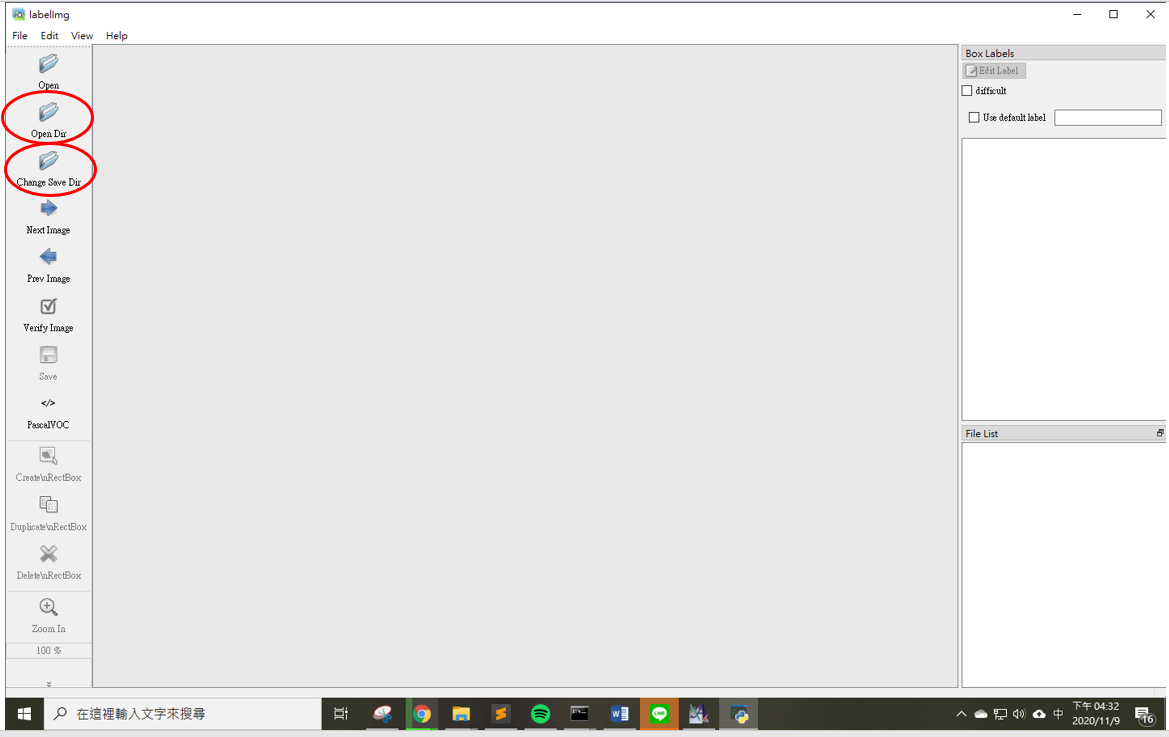


**安裝&操作 LabelImg：**

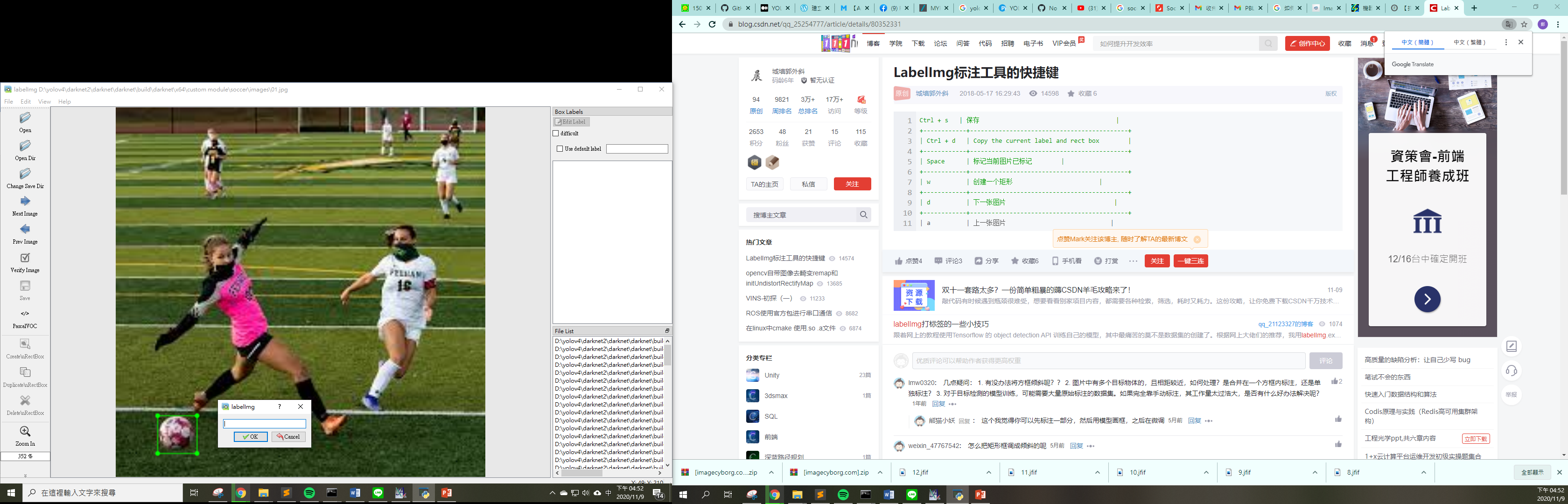
進入終端機輸入

pip install labelImg

安裝完後終端機輸入labelimg就會開啟以下視窗，按「Open Dir」、「Change Save Dir」選擇剛剛建立的images以及labels資料夾。



匯入後按下w拖曳範圍，選取後輸入名稱，這邊設soccer

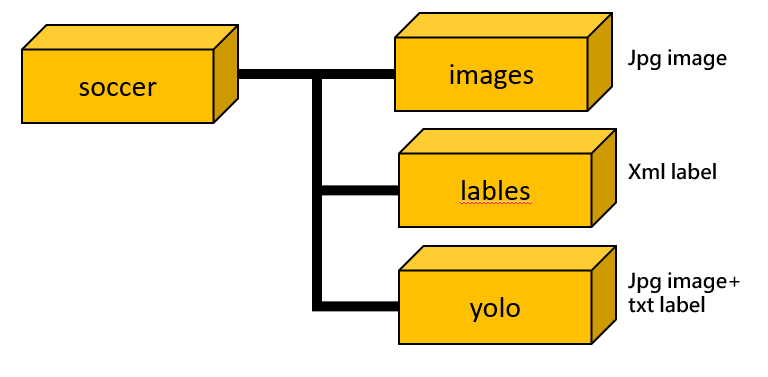


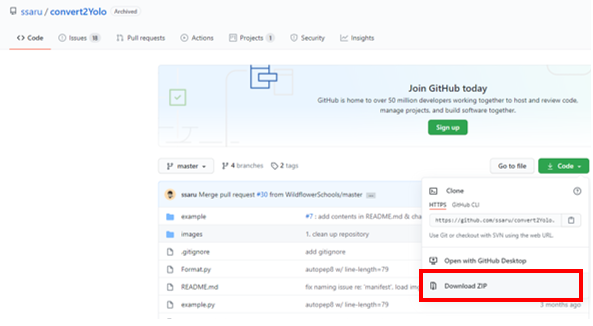
按ok後右邊會出現你設的名稱，按crtl+s儲存後即可按d下一張，直到所有標完(大概30~60mins)，也可進label資料夾是否有xml檔





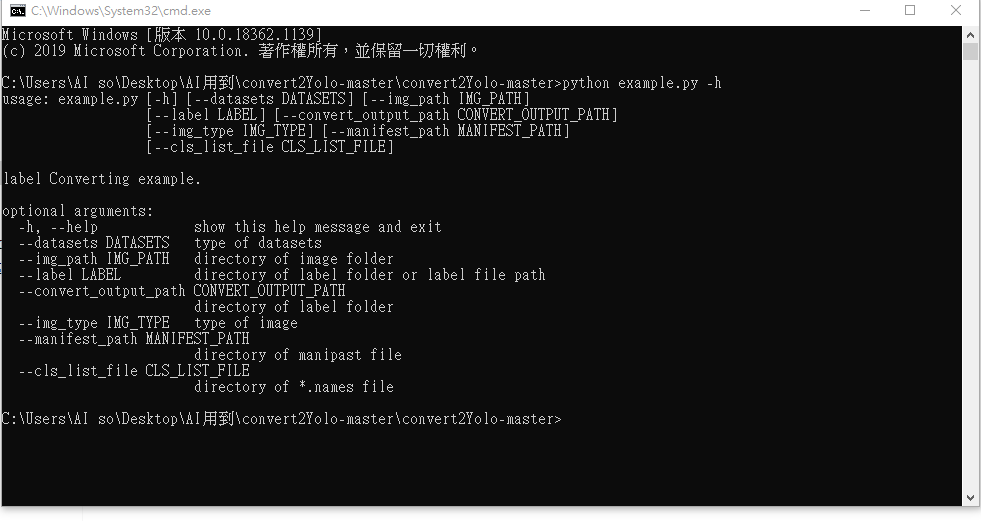
**Convert2Yolo：**

此步驟主要用來將xml檔轉換成txt檔，並生成訓練用txt，並在soccer下方建立一個yolo資料夾存放轉換出來的txt   
首先到[github](https://github.com/ssaru/convert2Yolo)下載整個檔案



解壓縮後放在你想要的位置，在該資料夾位置輸入python example.py -h

由視窗內可以看到需要補充的函數，而voc.names可看下個步驟內說明並線建立一個cfg的資料夾，再用任意編輯器開一個txt改名叫voc.names



--Dataset🡪要將voc(xml)格式轉換成txt

--img\_path🡪指到你image位置

--label🡪指到你label位置

--convert\_out\_put🡪轉換完要放置的資料夾(開一個放在soccer下)

--imagetype🡪”.jpg”

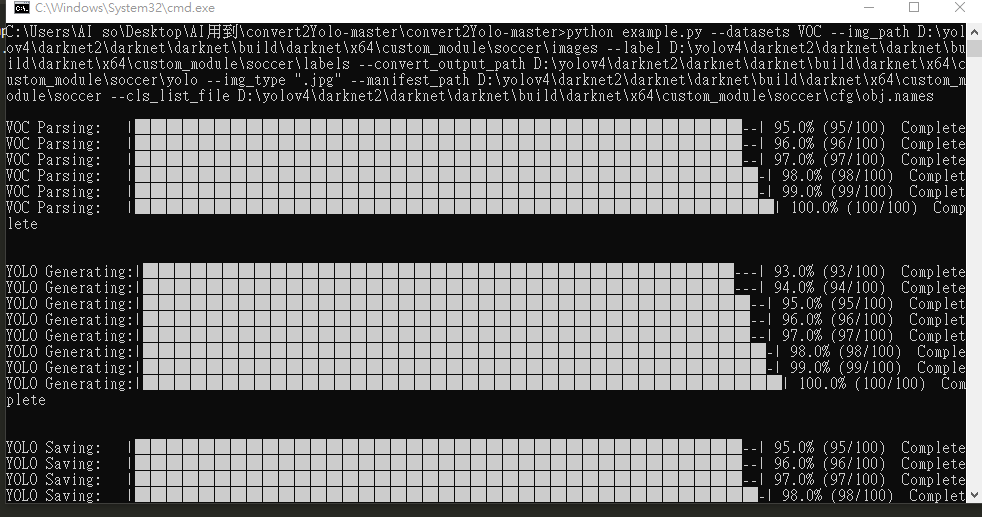
--manifest\_path🡪設你原本soccer就好

--cls\_list🡪到voc.names

所以我這輸入

python example.py --datasets VOC --img\_path D：\yolov4\darknet2\darknet\darknet\build\darknet\x64\custom\_module\soccer\images --label D：\yolov4\darknet2\darknet\darknet\build\darknet\x64\custom\_module\soccer\labels --convert\_output\_path D：\yolov4\darknet2\darknet\darknet\build\darknet\x64\custom\_module\soccer\yolo --img\_type ”.jpg” --manifest\_path D：\yolov4\darknet2\darknet\darknet\build\darknet\x64\custom\_module\soccer --cls\_list\_file D：\yolov4\darknet2\darknet\darknet\build\darknet\x64\custom\_module\soccer\cfg\obj.names

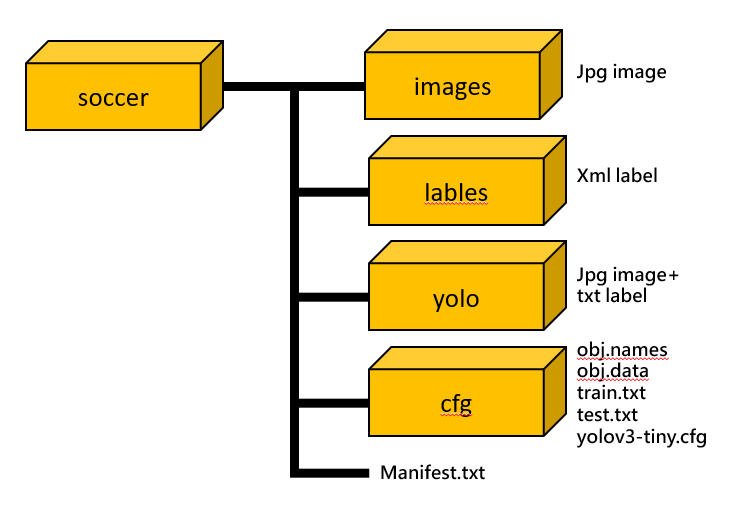
執行完得到以下視窗



**建立資料夾存放cfg結果：**

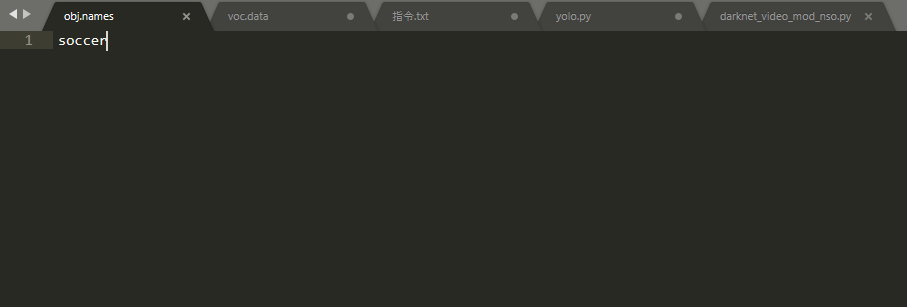
因YOLO的label檔是txt，每張圖片對應一個txt檔，且兩者較要放置於同一資料夾中，因此建資料夾名稱為yolo，將所有images及txt檔全部放置於此

而cfg內較複雜

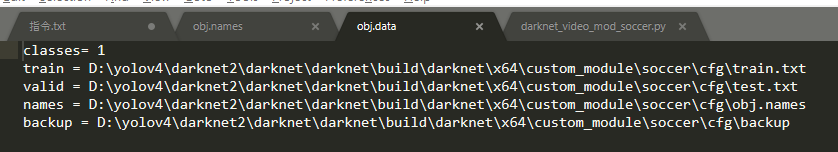


主要有5個檔案

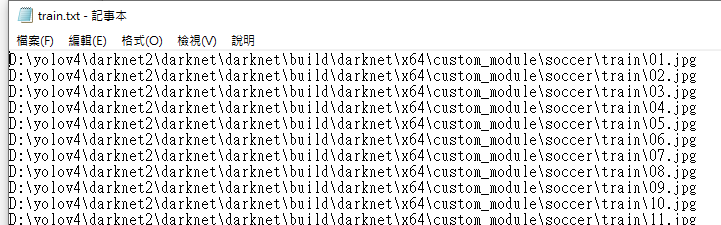
1. obj.names：此檔內容為label的列表，例如這裡只有soccer。



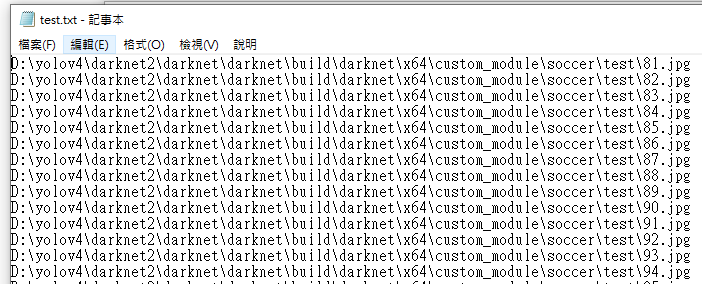
1. obj.data：定義label數目以及各個設定檔path。



1. triain.txt：所有images檔案名稱列表中的80%（或其它比例，可視需求變更），訓練時YOLO會依次讀取該檔內容取出相片進行訓練.。您可以手動或寫程式取出固定比例的列表放置於此檔案內容中。



1. test.txt：所有images檔案名稱列表中的20%（或其它比例，可視需求變更），訓練時YOLO會依次讀取該檔內容取出相片進行validation.。您可以手動或寫程式取出固定比例的列表放置於此檔案內容中。



1. cfg：這邊選擇你要使用的cfg檔，其實每個版本要修的東西大同小異，到D：\yolov4\darknet\cfg找你要的cfg，並複製到你的cfg資料夾

**如果您想訓練Tiny YOLO，請複製並修改yolov4-tiny-custom.cfg如下**

Line3： set batch=24 → using 24 images for every training step

Line 4： set subdivisions=8 → the batch will be divided by 8

Line 20： max\_batches=10000 → training rate

Line 22： steps=8000,9000 → training step

Line 212： set filters=(classes + 5)\*3   → in our case filters=18

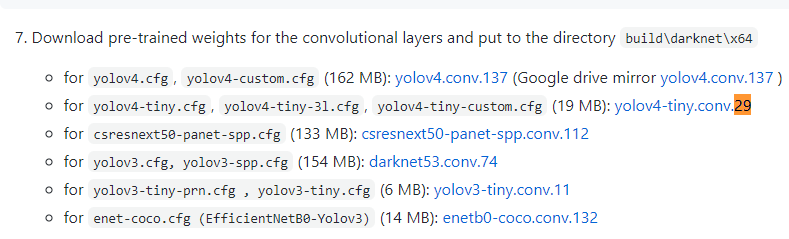
Line 220： set classes=1  →  the number of categories we want to detect

Line 263： set filters=(classes + 5)\*3  → in our case filters=18

Line 269： set classes=1   → the number of categories we want to detect

**下載權重檔：**

到作者[github](https://github.com/AlexeyAB/darknet" \l "how-to-train-to-detect-your-custom-objects)下載權重檔(yolov4-tiny.conv.29)並放在你的cfg資料夾內



**開始訓練!：**

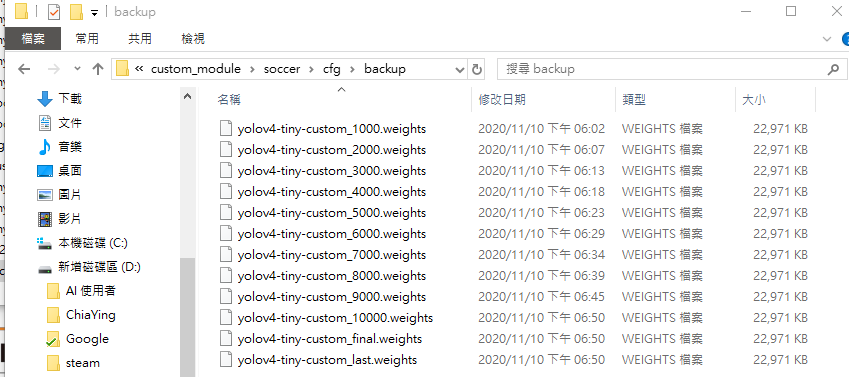
進cmd輸入

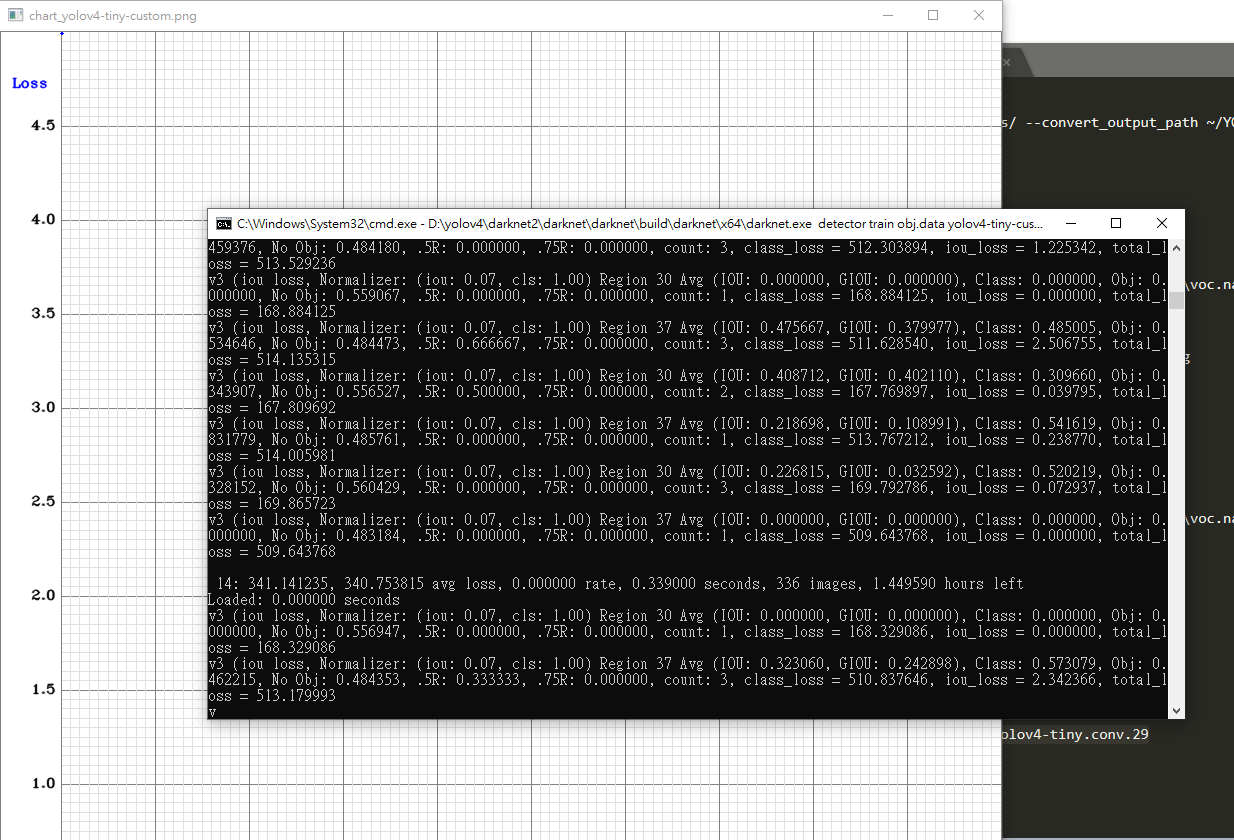
D:\yolov4\darknet2\darknet\darknet\build\darknet\x64\darknet.exe detector train obj.data yolov4-tiny-custom.cfg ./yolov4-tiny.conv.29

Or

D:\yolov4\darknet2\darknet\darknet\build\darknet\x64\darknet.exe detector train obj.data yolov4-custom.cfg ./yolov4.conv.137

螢幕會出現一個chart，，如果cmd不像下圖代表前面步驟有錯，請按crtl+c取消訓練，若成功等他跑完會在backup出現很多weight檔

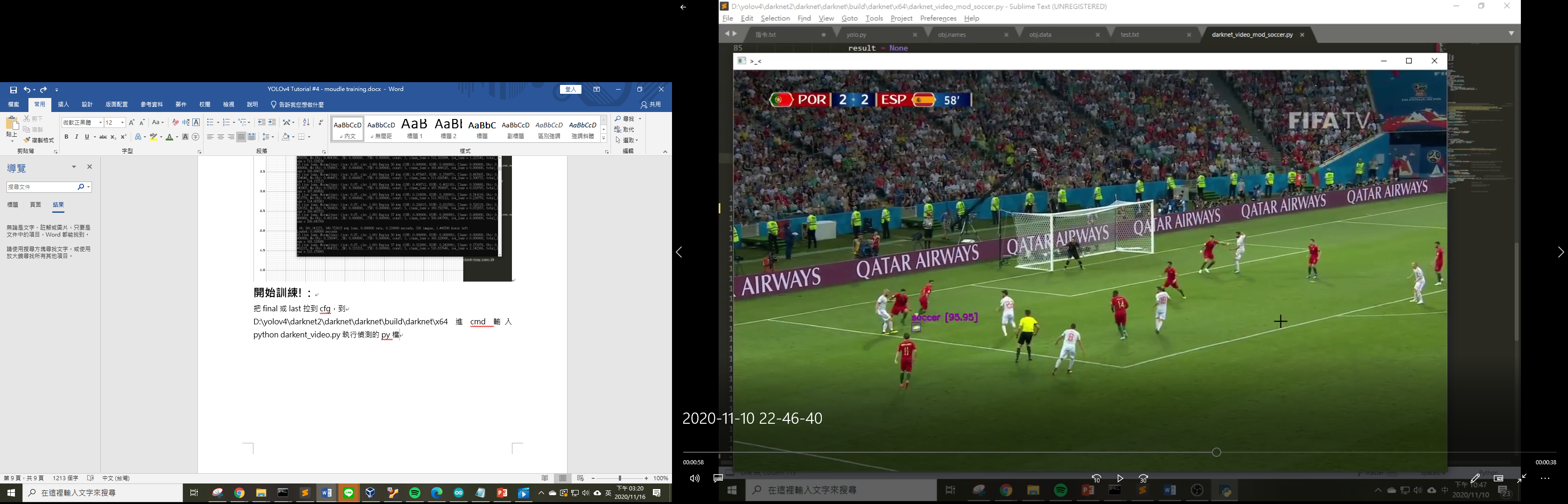




**開始偵測!：**

把final或last拉到cfg，到g

D:\yolov4\darknet2\darknet\darknet\build\darknet\x64進cmd輸入python darkent\_video.py執行偵測的py檔



**\*心得：**

我有試著將train的樣本數從80增加至150，錄影下來觀察但其實沒差很多，但如果是要做特定物品辨識，例如工廠內物件其實不太會失誤，因為像足球外在因素太多，而工廠內頂多是被遮蔽住。