環境安裝：

conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch

git clone https://github.com/Megvii-BaseDetection/YOLOX.git

cd YOLOX

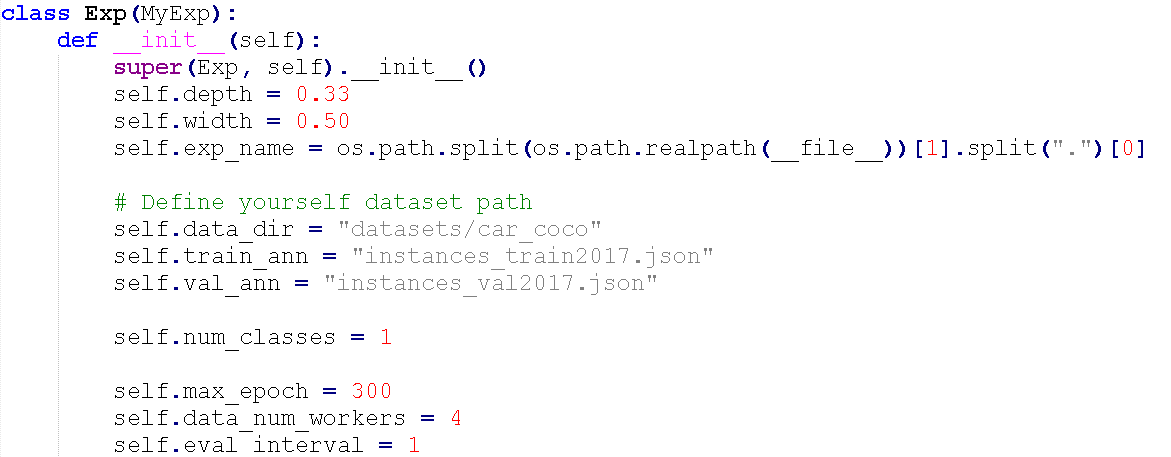
pip install -r requirements.txt

pip install -v -e .

資料預處理：

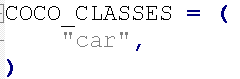
* Convert yolo format to coco format:
  1. 在yolo2coco.py目錄下新增datas資料夾
  2. 將train/val與train\_labels /val\_labels資料集移到yolo2coco.py目錄下
  3. 訓練集轉換： python yolo2coco.py -ap train\_labels -s datas --image-path train --json-name instances\_train2017.json
  4. 驗證集轉換： python yolo2coco.py -ap val\_labels -s datas --image-path val --json-name instances\_val2017.json
  5. 將instances\_train2017移到YOLOX\datasets\car\_coco\annotations下
  6. 將train/val命名成train2017/val2017並移到YOLOX\datasets\car\_coco下
* yolo2coco.py資料參考：<https://zhuanlan.zhihu.com/p/461488682>

訓練：

* 修改YOLOX\exps\example\custom\yolox\_s.py內的參數(如下圖所示)

← Classes設定成1

← Epoch自行調整

* 修改YOLOX\yolox\data\datasets\coco\_classes.py(如下圖)：
* 下載初始權重到YOLOX下(<https://github.com/Megvii-BaseDetection/YOLOX/releases/download/0.1.1rc0/yolox_s.pth>)
* 開始訓練：

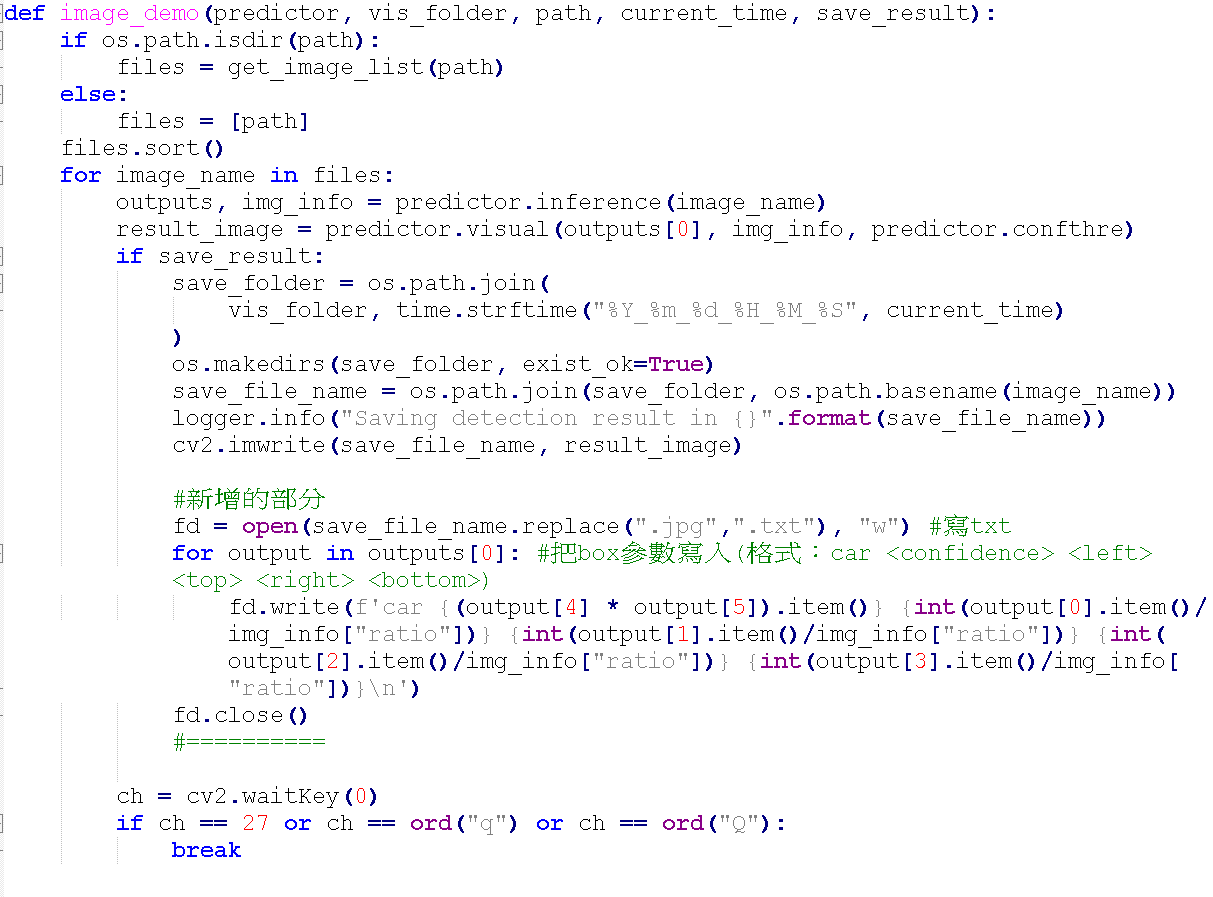
python tools/train.py -f exps/example/custom/yolox\_s.py -d 1 -b 24 --fp16 -o -c yolox\_s.pth

預測test資料集：

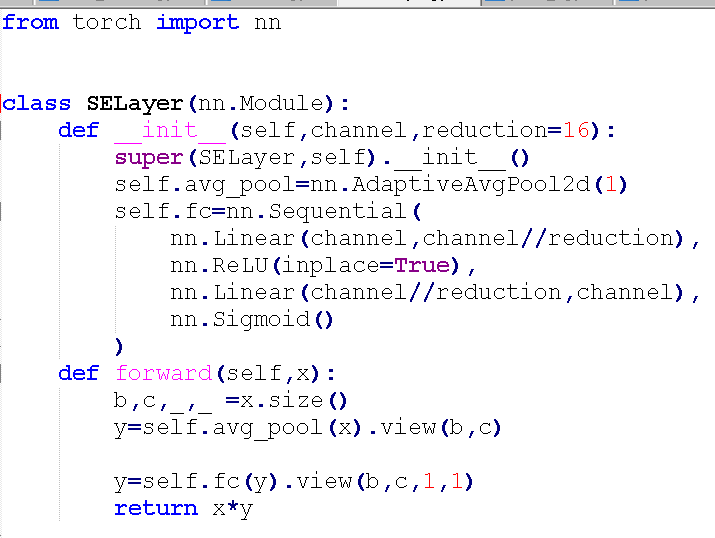
* 將test資料集移到YOLOX/datasets/car\_coco下
* python tools/demo.py image -f exps/example/custom/yolox\_s.py -c YOLOX\_outputs/yolox\_s/best\_ckpt.pth --path datasets/car\_coco/test/ --conf 0.25 --nms 0.5 --tsize 640 --save\_result --device gpu

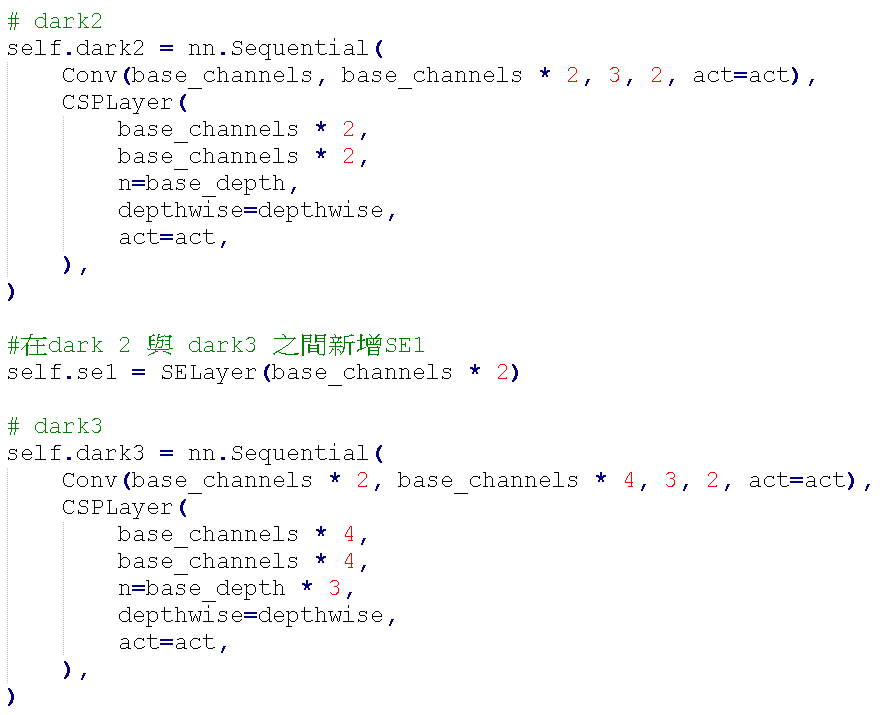
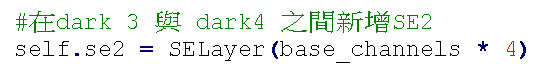
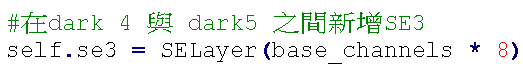
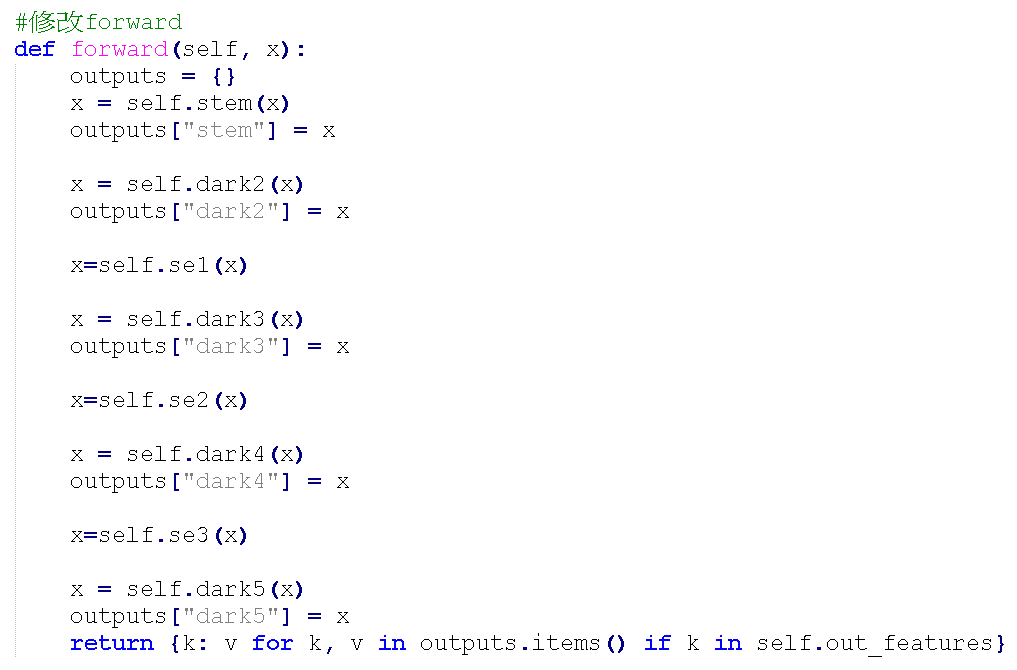
CODE解釋：

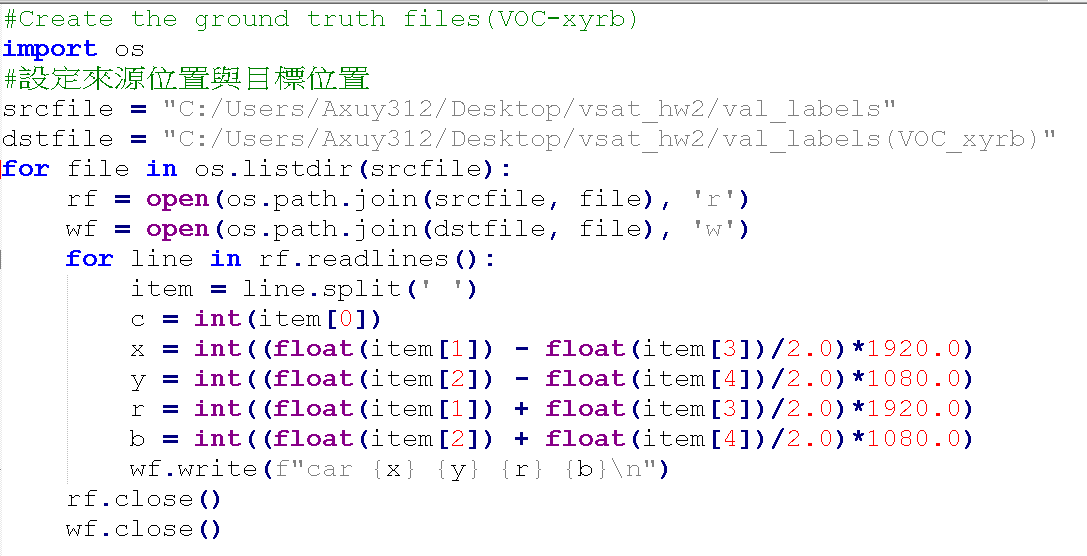
* 因為預測test資料集要產生box的座標與類別所以要修改Original\tools\demo.py(如下圖)
* txt格式：<class\_name> <confidence> <left> <top> <right> <bottom>



* 新增檔案為YOLOX\yolox\models\SELayer.py (如下圖)

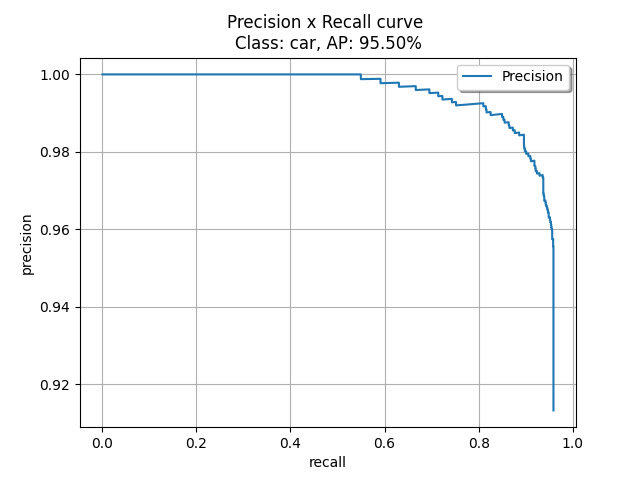
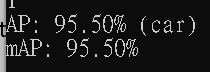


* 修改YOLOX\yolox\models\darknet.py (如下圖)    
* 將label從yolo format轉為car <left> <top> <right> <bottom> (yolo2voc.py如下圖)
* 計算mAP: python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np

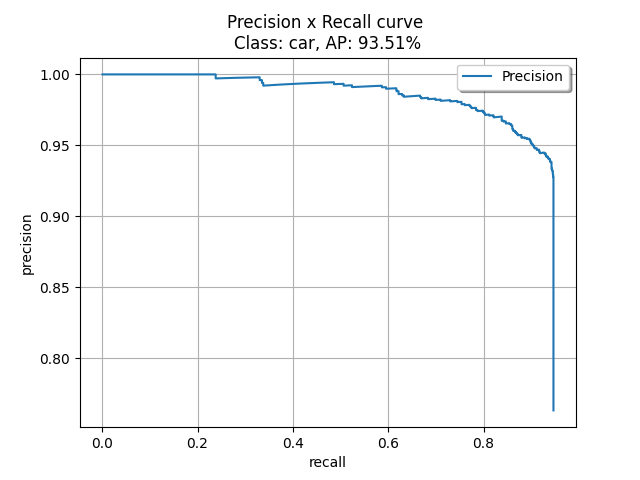
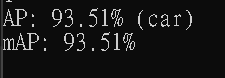


驗證集結果：

* python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np
* without SE(如下圖)



* with SE(如下圖)



問題與討論：

* 一開始要先知道有三種format: yolo, voc, coco
* 要計算mAP時要轉換txt的格式，所以需要trace demo.py並新增產生txt的檔案的code，並且還要將原先groundtruths txt的座標格式轉換成<left> <top> <right> <bottom>，最後才能使用github “Object-Detection-Metrics”來計算mAP
* 測試過沒有SE跑300 epoch可以達到95.5%的mAP、加一層SE跑50 epoch可達到92.22%的mAP、加三層SE跑300 epoch可達到93.51%的mAP

檔案位置：

* Without SE Model: Code\Original\YOLOX\YOLOX\_outputs\yolox\_s\best\_ckpt.pth
* With SE Model(三層SE、300 epoch): Code\SE\YOLOX\YOLOX\_outputs\yolox\_s\best\_ckpt.pth
* With SE Model(一層SE、50 epoch):

Code\SE\YOLOX\YOLOX\_outputs\yolox\_s\ best\_ckpt(92.22%).pth

* 因為E3檔案大小限制，所以zip中沒有放資料集(jpg檔)