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)

```
class Exp(MyExp):
    def __init__(self):
        super(Exp, self).__init__()
        self.depth = 0.33
        self.width = 0.50
        self.exp_name = os.path.split(os.path.realpath(__file__))[1].split(".")[0]

        # Define yourself dataset path
        self.data_dir = "datasets/car_coco"
        self.train_ann = "instances_train2017.json"
        self.val_ann = "instances_val2017.json"

        self.num_classes = 1  ← Classes 設定成 1

        self.max_epoch = 300  ← Epoch 自行調整
        self.data_num_workers = 4
        self.eval_interval = 1
```

- 修改 YOLOX\yolox\data\datasets\coco_classes.py(如下圖)：

```
COCO_CLASSES = (
    "car",
)
```

- 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>

```
def image_demo(predictor, vis_folder, path, current_time, save_result):
    if os.path.isdir(path):
        files = get_image_list(path)
    else:
        files = [path]
    files.sort()
    for image_name in files:
        outputs, img_info = predictor.inference(image_name)
        result_image = predictor.visual(outputs[0], img_info, predictor.confthre)
        if save_result:
            save_folder = os.path.join(
                vis_folder, time.strftime("%Y_%m_%d_%H_%M_%S", current_time)
            )
            os.makedirs(save_folder, exist_ok=True)
            save_file_name = os.path.join(save_folder, os.path.basename(image_name))
            logger.info("Saving detection result in {}".format(save_file_name))
            cv2.imwrite(save_file_name, result_image)

            #新增的部分
            fd = open(save_file_name.replace(".jpg",".txt"), "w") #寫txt
            for output in outputs[0]: #把box參數寫入(格式：car <confidence> <left>
            <top> <right> <bottom>)
                fd.write(f'car {(output[4] * output[5]).item()} {int(output[0].item()/
                img_info["ratio"])} {int(output[1].item()/img_info["ratio"])} {int(
                output[2].item()/img_info["ratio"])} {int(output[3].item()/img_info[
                "ratio"])}\n')
            fd.close()
            #==========

        ch = cv2.waitKey(0)
        if ch == 27 or ch == ord("q") or ch == ord("Q"):
            break
```

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

```
from torch import nn


class SELayer(nn.Module):
    def __init__(self,channel,reduction=16):
        super(SELayer,self).__init__()
        self.avg_pool=nn.AdaptiveAvgPool2d(1)
        self.fc=nn.Sequential(
            nn.Linear(channel,channel//reduction),
            nn.ReLU(inplace=True),
            nn.Linear(channel//reduction,channel),
            nn.Sigmoid()
        )
    def forward(self,x):
        b,c,_,_ =x.size()
        y=self.avg_pool(x).view(b,c)

        y=self.fc(y).view(b,c,1,1)
        return x*y
```

- 修改 YOLOX\yolox\models\darknet.py (如下圖)

```
        # dark2
        self.dark2 = nn.Sequential(
            Conv(base_channels, base_channels * 2, 3, 2, act=act),
            CSPLayer(
                base_channels * 2,
                base_channels * 2,
                n=base_depth,
                depthwise=depthwise,
                act=act,
            ),
        )

        #在dark 2 與 dark3 之間新增SE1
        self.se1 = SELayer(base_channels * 2)

        # dark3
        self.dark3 = nn.Sequential(
            Conv(base_channels * 2, base_channels * 4, 3, 2, act=act),
            CSPLayer(
                base_channels * 4,
                base_channels * 4,
                n=base_depth * 3,
                depthwise=depthwise,
                act=act,
            ),
        )

        #在dark 3 與 dark4 之間新增SE2
        self.se2 = SELayer(base_channels * 4)
```

```
#在dark 4 與 dark5 之間新增SE3
self.se3 = SELayer(base_channels * 8)


#修改forward
def forward(self, x):
    outputs = {}
    x = self.stem(x)
    outputs["stem"] = x

    x = self.dark2(x)
    outputs["dark2"] = x

    x=self.se1(x)

    x = self.dark3(x)
    outputs["dark3"] = x

    x=self.se2(x)

    x = self.dark4(x)
    outputs["dark4"] = x

    x=self.se3(x)

    x = self.dark5(x)
    outputs["dark5"] = x
    return {k: v for k, v in outputs.items() if k in self.out_features}
```

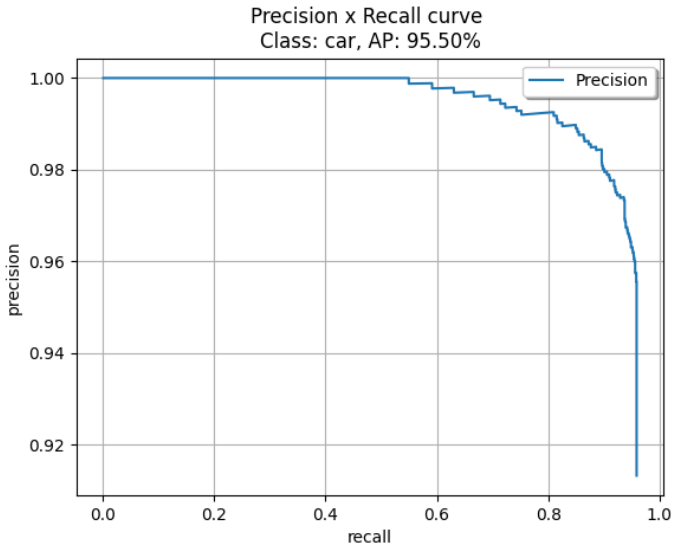
- 將 label 從 yolo format 轉為 car <left> <top> <right> <bottom> (yolo2voc.py 如下圖)
- 計算 mAP: python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np

```
#Create the ground truth files(VOC-xyrb)
import os
#設定來源位置與目標位置
srcfile = "C:/Users/Axuy312/Desktop/vsat_hw2/val_labels"
dstfile = "C:/Users/Axuy312/Desktop/vsat_hw2/val_labels(VOC_xyrb)"
for file in os.listdir(srcfile):
    rf = open(os.path.join(srcfile, file), 'r')
    wf = open(os.path.join(dstfile, file), 'w')
    for line in rf.readlines():
        item = line.split(' ')
        c = int(item[0])
        x = int((float(item[1]) - float(item[3])/2.0)*1920.0)
        y = int((float(item[2]) - float(item[4])/2.0)*1080.0)
        r = int((float(item[1]) + float(item[3])/2.0)*1920.0)
        b = int((float(item[2]) + float(item[4])/2.0)*1080.0)
        wf.write(f"car {x} {y} {r} {b}\n")
    rf.close()
    wf.close()
```

驗證集結果：

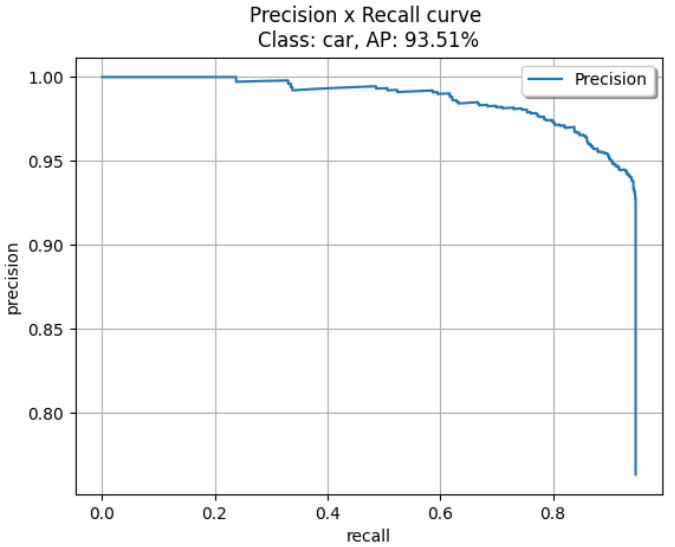
- python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np
- without SE(如下圖)

```
AP: 95.50% (car)
mAP: 95.50%
```



- with SE(如下圖)

```
AP: 93.51% (car)
mAP: 93.51%
```



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)