電腦視覺實務與深度學習

作業三

M11102137 黃科皓

1. Report the performance of your **trained source model** on the **source validation set**

   Select Model: ConfMix - https://arxiv.org/abs/2210.11539

   Train 160 epoch (0~159) with img_size=1280, hyp=hyp.scratch-high.yaml

   a. mAP@[50:5:95], mAP@50, mAP@75

   Source validation set:

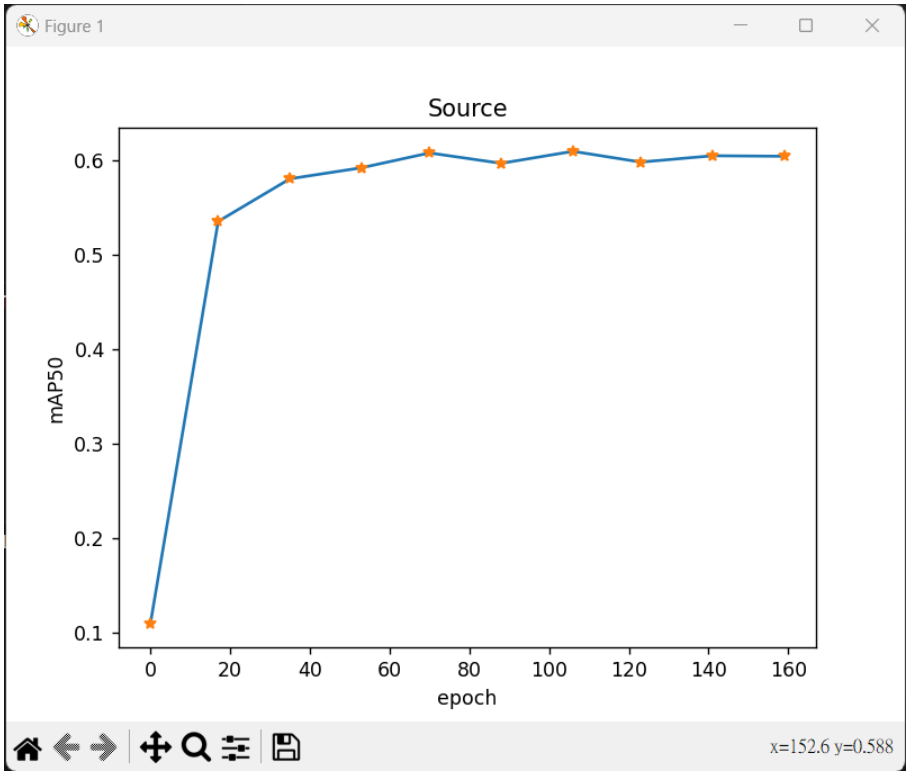
   source.pt (YOLOv5 val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.604	0.462	0.426
person	0.69	0.442	0.426
car	0.82	0.674	0.629
truck	0.501	0.393	0.383
bus	0.644	0.624	0.53
rider	0.648	0.492	0.439
motorcycle	0.551	0.358	0.319
bicycle	0.531	0.313	0.305
train	0.447	0.403	0.376

   source.pt (check_your_prediction_valid.py):

	mAP50	mAP75	mAP[50:5:95]
all	0.6026	0.4557	0.4224

   b. mAP50 curve (YOLOv5 results.csv) (img_size=1280):



2. Report the performance of your **trained source model** on the **target validation set (w/o any adaptations)**

   Target validation set:

   source.pt (YOLOv5 uda_val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.485	0.386	0.355
person	0.604	0.407	0.389
car	0.718	0.601	0.56
truck	0.308	0.269	0.249
bus	0.439	0.438	0.378
rider	0.562	0.441	0.397
motorcycle	0.462	0.302	0.286
bicycle	0.433	0.277	0.261
train	0.356	0.356	0.32

   source.pt (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.4816	0.3819	0.3510

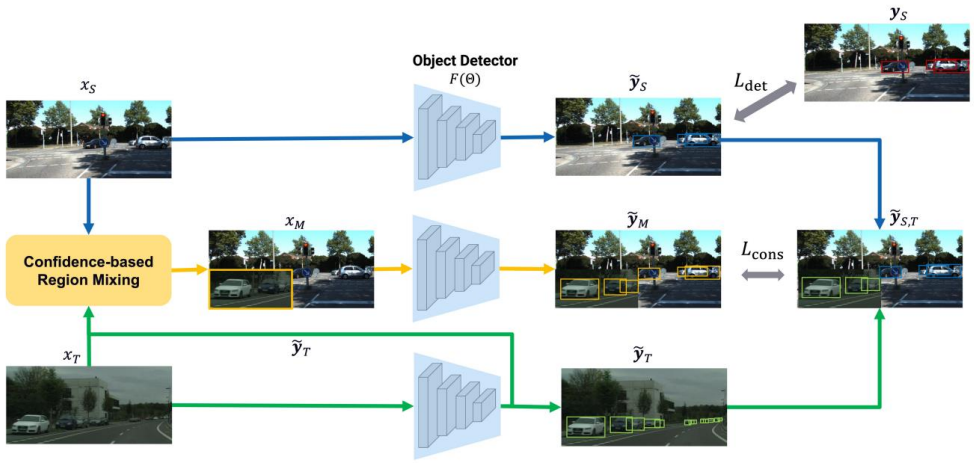
3. Please provide a introduction to the **two domain adaptation methods** you used.

A. ConfMix:

ConfMix - https://arxiv.org/abs/2210.11539

UDA 用在 object detection 是將在 source domain 上訓練的模型適應到新的 target domain，而該 target domain 上沒有 label 可以使用。與傳統方法不同， ConfMix 使用基於 region-level detection confidence 的 sample mixing strategy 方法，用於 adaptive object detector learning。將 target sample 中最高 confidence 的局部區域與 source image 進行混合，並使用額外的 consistency loss term，逐漸適應 target data distribution。

Architecture:



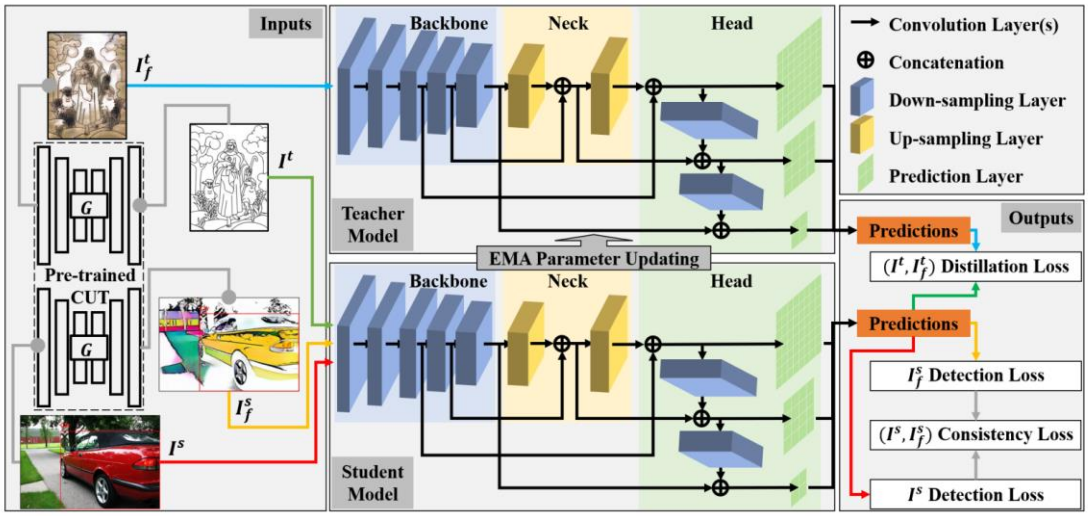
ConfMix 以 YOLOv5 做為訓練模型，將有 label 的 image 用一般 YOLO 的方式做 supervised learning。而沒有 label 的資料先 pridict 出結果，把 confidence 高的區域做為 pseudo label 做 training。L_{cons} 為 mixed sample 的 loss，L_{det} 為 labeled sample 的 loss。

B. SSDA-YOLO:

SSDA-YOLO - https://arxiv.org/abs/2211.02213v2

SSDA-YOLO 利用了 labeled 和 unlabeled data 來訓練模型。該方法基於 YOLO 目標檢測框架，該框架具有快速和高效的特點。SSDA-YOLO 通過兩個關鍵步驟來實現跨領域目標檢測。首先，它使用 labeled 的 source domain data 訓練初始的目標檢測模型。然後使用 unlabeled 的 target domain data 做 adaptive learing，使其能夠適應 target domain 的特徵分佈。

Architecture:



SSDA-YOLO 將 real source domain data 用 CUT 轉換成 fake target domain data，以及將 real target domain data 用 CUT 轉換成 fake source domain data。把 real target domain data 輸入 teacher model 中，把 real source domain data、fake source domain data 和 fake target domain data 輸入 student model 中。使用 teacher student learning 的方式來做學習，將 real target domain data 得預測結果和 fake target domain data 的預測結果做 distillation loss。Real source domain data 和 fake source domain data 的預測結果使用 label 算 detection loss，把兩者的 detection loss 結合做為 consistency loss。

4. Please compare the two methods and describe their respective **advantages and disadvantages**.

	ConfMix	**SSDA-YOLO**
Advantage	1. 擴展數據集：pseudol abel 可以利用 target domain unlabeled data 擴展 source domain dataset，從而增加訓練數據的多樣性。 2. 降低 label 成本：相比於需要手動 label target dataset 的方法，pseudo label 不需要人工 label，因此可以節省大量的成本。	1. 知識轉移：teacher model 可以傳遞在 source domain 上學習到的知識給 student model，幫助 student model 更好地適應 target domain。 2. 適應性學習：Teacher student model 可以根據 target domain 的特徵分佈調整模型的參數，實現更好的適應效果。
disadvantage	1. Pseudo label 可靠性：由於預測結果可能存在錯誤，pseudo label 的可靠性成為一個關鍵問題。如果 pseudo label 不準確，將會導致模型學習到不正確的知識。 2. Domain difference：Pseudo label 並未充分考慮 source domian 和 target domain 之間的 domain difference。在 target domain 存在較大差異的情況下，pseudo label 可能無法取得良好的適應效果。	1. 依賴 teacher model：Teacher model 的品質和準確性對 student model 的表現有很大的影響。如果 teacher model 不夠準確，將會影響 student model 的性能。 2. 過程複雜：teacher student model 需要設計適合的略和損失函數，這需要一定的專業知識和調參過程。

5. Report the performance of the **adapted model** on the **target validation set**

mAP50 increase from 0.4816 to 0.5172, magnitude = (0.5172- 0.4816)/ 0.4816 = 7.392%

Train 30 epoch (0~29) img_size=1280

a. mAP@[50:5:95], mAP@50, mAP@75

source.pt(YOLOv5 uda_val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.485	0.386	0.355
person	0.604	0.407	0.389
car	0.718	0.601	0.56
truck	0.308	0.269	0.249
bus	0.439	0.438	0.378
rider	0.562	0.441	0.397
motorcycle	0.462	0.302	0.286
bicycle	0.433	0.277	0.261
train	0.356	0.356	0.32

source.pt (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.4816	0.3819	0.3510

epoch10.pt(YOLOv5 uda_val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.467	0.34	0.311
person	0.611	0.398	0.373
car	0.764	0.599	0.553
truck	0.31	0.272	0.242
bus	0.438	0.424	0.357
rider	0.548	0.411	0.351
motorcycle	0.5	0.281	0.274
bicycle	0.438	0.228	0.239
train	0.126	0.11	0.0977

epoch10.pt (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.4647	0.3368	0.3091

epoch20.pt(YOLOv5 uda_val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.517	0.399	0.367
person	0.637	0.418	0.392
car	0.78	0.627	0.585
truck	0.408	0.352	0.334
bus	0.566	0.536	0.468
rider	0.58	0.459	0.39
motorcycle	0.412	0.261	0.258
bicycle	0.448	0.234	0.259
train	0.308	0.308	0.253

epoch20.pt (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.5138	0.3953	0.3643

epoch29.pt(YOLOv5 uda_val.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.522	0.402	0.369
person	0.638	0.436	0.407
car	0.784	0.641	0.598
truck	0.424	0.339	0.329
bus	0.547	0.511	0.452
rider	0.576	0.459	0.397
motorcycle	0.443	0.337	0.287
bicycle	0.462	0.258	0.268
train	0.303	0.231	0.213

Epoch29.pt (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.5172	0.3963	0.3637

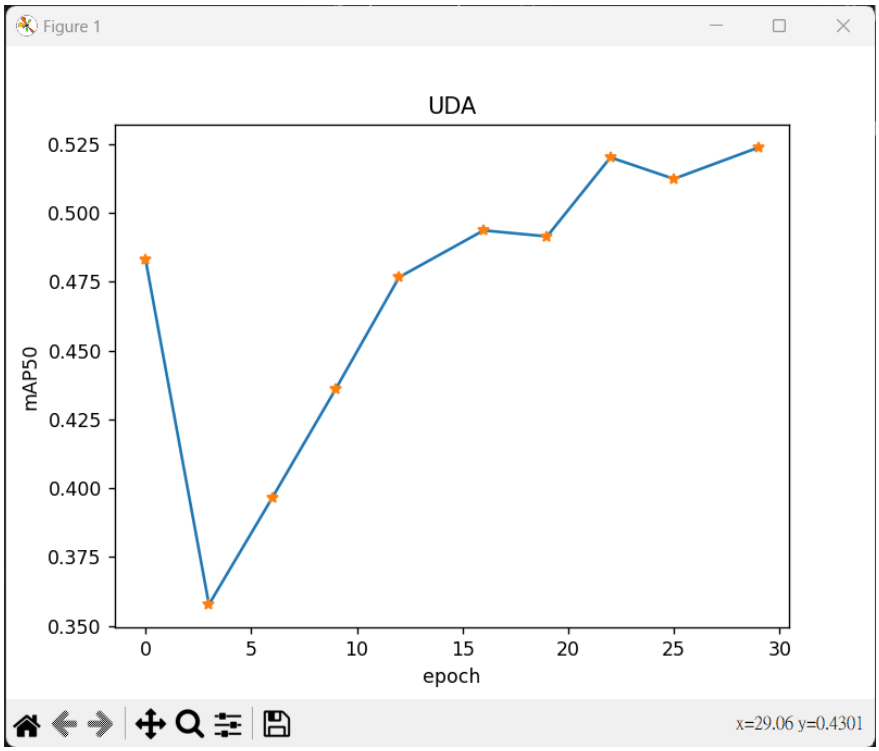
best: epoch29.pt(YOLOv5 uda_val.py) (change img_size to 2000):

	mAP50	mAP75	mAP[50:5:95]
all	0.55	0.424	0.393
person	0.704	0.522	0.473
car	0.828	0.708	0.648
truck	0.394	0.29	0.288
bus	0.546	0.491	0.431
rider	0.625	0.515	0.454
motorcycle	0.504	0.327	0.313
bicycle	0.493	0.302	0.303
train	0.304	0.236	0.235

best: epoch29.pt(check_your_prediction_valid.py) (change img_size to 2000):

	mAP50	mAP75	mAP[50:5:95]
all	0.5464	0.4188	0.3907

b. mAP50 curve(YOLOv5 results.csv):



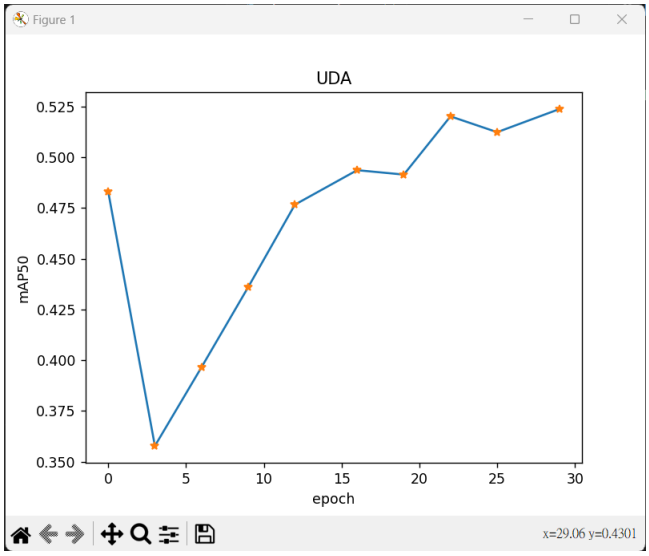
7. Please compare the final mAP50 of the adapted model trained from the following two different initial weights.

下面的結果為使用預設 yolov5m6 model 對 unlabeled fog data 訓練和使用先對 org data train 做 pre-train 的 yolov5m6 model 對 unlabeled fog data 訓練。從兩次的結果可以看出有 pre-train 的 model 一開始的 performance 會比較好，因為對 org data 和 fog data 有一定的關聯性。但是在做很多 epoch 的 training 後，performance 的結果差不多。這是因為 ConfMix 是使用 labeled org data 和 unlabeled fog data 做 training，所以做 domain adaptive training 時也同時對 labeled org data 做 training，導致兩個 model 的差異會逐漸縮小。

With training source data:

epoch29.pt (check_your_prediction_valid.py) (img_size=1280):

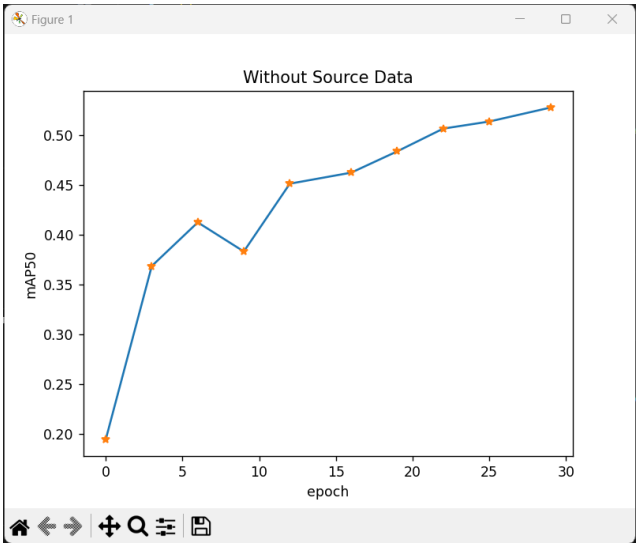
	mAP50	mAP75	mAP[50:5:95]
all	0.5172	0.3963	0.3637



Without training source data:

epoch29 (check_your_prediction_valid.py) (img_size=1280):

	mAP50	mAP75	mAP[50:5:95]
all	0.5221	0.3837	0.3580



References:

1. Mattolin, Giulio, et al. "ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023.
2. Zhou, Huayi, Fei Jiang, and Hongtao Lu. "SSDA-YOLO: Semi-supervised domain adaptive YOLO for cross-domain object detection." *Computer Vision and Image Understanding* 229 (2023): 103649.
3. Oza, Poojan, et al. "Unsupervised domain adaptation of object detectors: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
4. https://github.com/giuliomattolin/ConfMix
5. https://github.com/hnuzhy/SSDA-YOLO