Computer Vision Practice with Deep Learning homework 3

生醫電資所 R11945070 陳光唯

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

我所選擇的 Source model 是 YOLOv5 (You Only Look Once)模型，是一種經常用於 object detection 的模型，Pretrain on COCO datasets

- **Source model: YOLOv5: yolov5x (Pretrain on COCO datasets)/batch size=8 / epochs=170/ optimizer=AdamW**

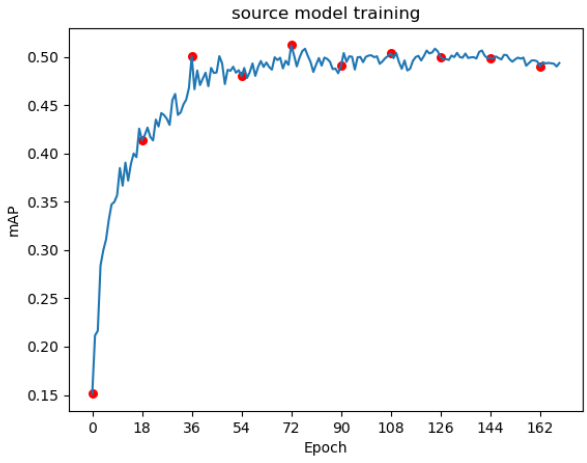
- **mAP@[50:5:95], mAP@50, mAP@75**
  YOLOv5 原生 mAP 列表:

Class	Precision	Recall	mAP50	mAP75	mAP[50:5:95]
All	0.722	0.478	0.507	0.343	0.332
Person	0.76	0.515	0.573	0.301	0.318
Car	0.852	0.688	0.755	0.576	0.546
Truck	0.586	0.4	0.415	0.293	0.297
Bus	0.922	0.516	0.616	0.5	0.457
Rider	0.756	0.472	0.53	0.351	0.319
Motorcycle	0.611	0.517	0.437	0.217	0.244
Bicycle	0.711	0.449	0.454	0.231	0.244
Train	0.576	0.286	0.276	0.274	0.235

Result of check_your_prediction_valid.py:

Class	mAP_small	mAP_medium	mAP_large
All	0.0328	0.2456	0.5579
	mAP50	mAP75	mAP[50:5:95]
	0.5037	0.3394	0.3300

- **mAP curve**



從上圖可以看出 YOLOv5 在不同物件的偵測上都算是非常優秀 ，收斂速度也快，而細部分類也有如 Train 分數較低，可能是因為 data imbalance (相較其他種類在訓練資料出現數量較少)導致的，雖然有嘗試進行 class weight 的正規化，但可能效果有限。

直接使用訓練結果測試 org 的 validation sets 效果非常好，平均 mAP50 有 0.5037 的成果，雖然可能因為 datasets 數量太少，依然達不到官網公告的 mAP50 那麼高分。

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

- **Source model: YOLOv5: yolov5x (Pretrain on COCO datasets)/batch size=8 / epochs=170/ optimizer=AdamW**

- **mAP@[50:5:95], mAP@50, mAP@75**
  YOLOv5 原生 mAP 列表:

Class	Precision	Recall	mAP50	mAP75	mAP[50:5:95]
All	0.678	0.397	0.418	0.294	0.276
Person	0.774	0.424	0.5	0.286	0.283
Car	0.88	0.546	0.659	0.512	0.486
Truck	0.439	0.375	0.299	0.242	0.225
Bus	0.932	0.37	0.447	0.395	0.345
Rider	0.669	0.471	0.488	0.352	0.317
Motorcycle	0.686	0.455	0.442	0.256	0.245
Bicycle	0.628	0.381	0.395	0.203	0.216
Train	0.415	0.154	0.11	0.10509	0.0911

- **Result of check_your_prediction_valid.py:**

Class	mAP_small	mAP_medium	mAP_large
All	0.0118	0.1765	0.5206
	mAP50	mAP75	mAP[50:5:95]
	0.4161	0.2896	0.2741

從上圖可以看出直接使用訓練結果測試 fog 的 validation sets 效果仍舊有一定的水平，但 mAP50 依然確實受到了 fog 的氣候影響而降低了許多，降至了 0.4161 的狀態，原本表現就不好的 Train 類別也明顯降至了非常低的地步: mAP50=0.11，因此要使用 UDA 的技術來救回受到 fog 影響的分數。

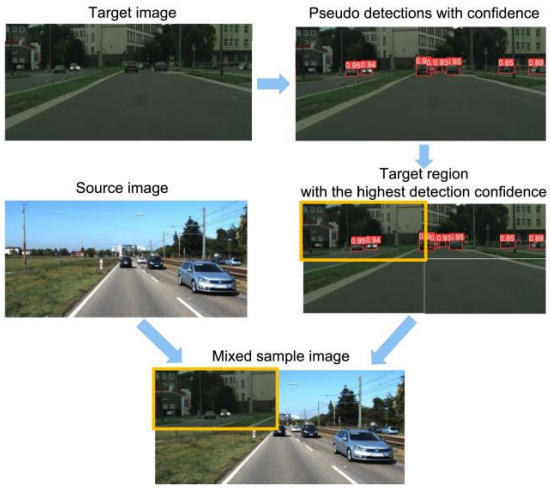
3. Please provide a introduction to the two domain adaptation methods you used

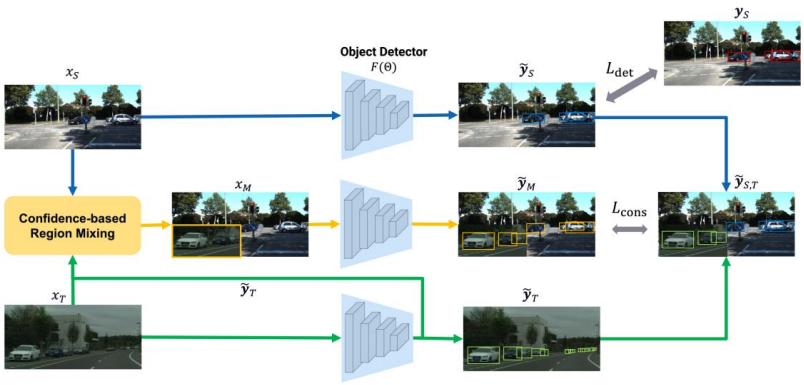
我使用 ConfMix 對我的 YOLOv5 source model 實際進行了 domain adaption，而 SSDA-YOLO 則是另一實現 domain adaption 的技術，我是藉由文獻去了解他如何運作，它們各自所使用的方法和架構各不相同，以下將簡短說明一下這兩個技術。

- **方法一: ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing:**

  當在訓練過程中未遇到的 Domain 不同的圖像上進行測試時，model 的性能往往會嚴重下降，這是由於 domain 的漂移所引起的。為了解決這個問題， Unsupervised Domain Adaptation 技術就顯得非常重要，還可以省去繁瑣的手動 labels 步驟。

  絕大多數的 UDA 方法是用對抗性訓練的方式，跟過去的 Unsupervised Domain Adaptation (UDA)方法不同，ConfMix 是第一種基於 region-level 檢測 confidence，並利用 sample mixing 來達到 UDA 的新方法。藉由將信心度最高的 pseudo detections 相對應的 target sample 的 local 區域與 source image 進行混合，並使用額外的 consistency loss 項，來逐漸適應 target data 的分佈，簡單來說就是通過將模型最有信心的 target 圖像區域與 source 圖像結合，人工生成樣本，並通過一致性 loss，使生成的圖像之間具有一致的預測。為了定義清楚各區域的信心分數，作者利用每個 pseudo detection 的信心分數來考慮 detector-dependent confidence 的信心程度，同時也考慮了 bounding box 的不確定性。此外作者也提出了一種新的 pseudo labelling 方式，通過使用隨訓練過程中變化的 loose to strict 的信心度指標逐漸過濾 pseudo target detections。此方法通過在三個數據集上的大量實驗(Cityscapes → FoggyCityscapes，Sim10K → Cityscapes 和 KITTI → Cityscapes)，在其中兩個數據集中達到了目前最好的性能，在另一個數據集上也達到了逼近 supervised target model 的表現，因此是一個非常有效的 UDA 方式，下圖及是 ConfMix 的實作流程。





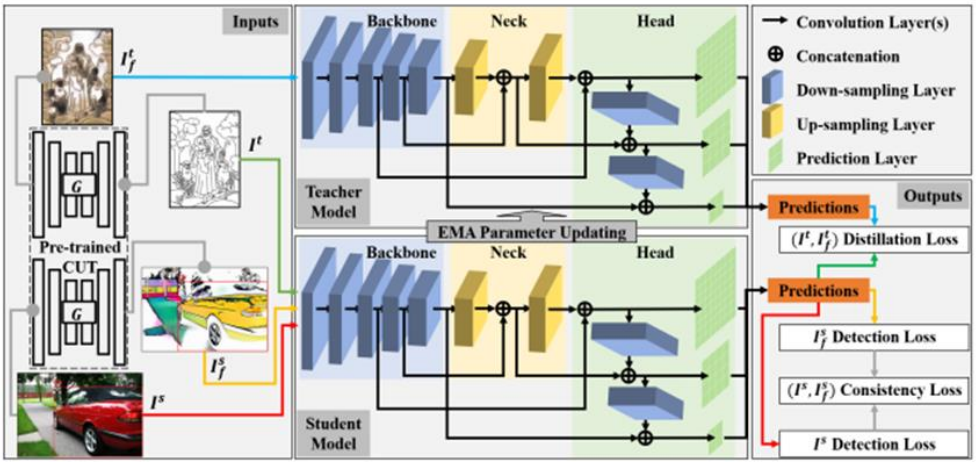
下圖則是實際的 mAP50 在經過 ConfMix 後的提升量範例:

				Sim10K→ Cityscapes	KITTI→ Cityscapes
Method	Detector	Backbone	Pretrained	mAP	mAP
Source only	YOLOv5	CSP-Darknet53	No	33.9	21.7
ConfMix (Ours)	YOLOv5	CSP-Darknet53	No	46.2	50.1
Oracle	YOLOv5	CSP-Darknet53	No	64.1	64.1
Source only	YOLOv5	CSP-Darknet53	COCO	49.5	39.9
ConfMix (Ours)	YOLOv5	CSP-Darknet53	COCO	56.3	52.2
Oracle	YOLOv5	CSP-Darknet53	COCO	70.3	70.3

- **方法二: SSDA-YOLO: Semi-supervised Domain Adaptive YOLO for Cross-Domain Object Detection:**

  CNN 準確且快速，但這些方法的高準確度大多僅限於訓練集原本的 domain，當在完全不同的 target domain 下進行測試時，往往會出現顯著的性能下降，一階段 object detector 的 YOLO 系列，在保持相當準確度的同時，幾乎可以達到實時檢測，這使它們在自動駕駛和動作識別等應用上非常重要。過去的 Faster R-CNN 方法在 domain adaption 佔研究的主導地位，但這可能會限制整體性能。儘管先進的強大檢測器 YOLOv5 目前展示出了高性能和耗時平衡的優勢，因此使用 YOLOv5 來進行 domain adaption 似乎是一個有效的方式。

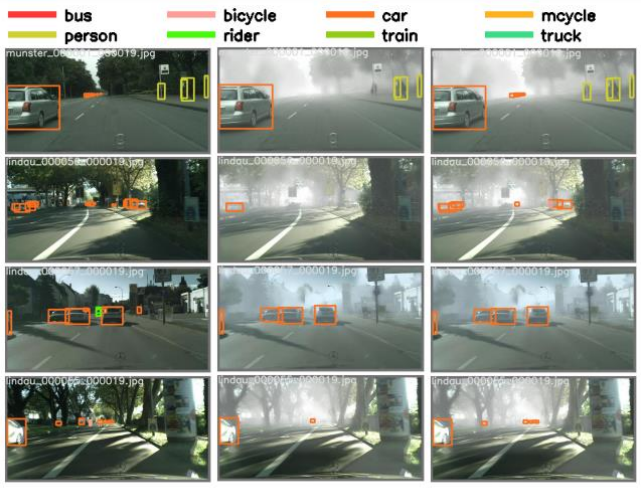
  此篇 paper 的構想主要源自於大多數現有的 Domain adaptive object detection (DAOD)方法都受到過時且計算複雜的兩階段 Faster R-CNN 的影響，實用價值較低，因此作者提出了一種基於半監督域自適應 YOLO（SSDA-YOLO）的新方法，通過將快速的單階段檢測器 YOLOv5 與域自適應相結合，以提高跨域檢測性能。主要方式運用 Mean Teacher 模型適應 knowledge distillation framework，以幫助 student 模型獲得未標記 target domain 的 instance-level features，同時還利用 GAN 來進行場景風格轉換在不同 domain 中交叉生成假圖像，以補償 image-level 的差異，並透過 consistency loss，進一步 align cross-domain 的預測。利用包含 PascalVOC、Clipart1k、Cityscapes 和 Foggy Cityscapes 在內的多項基準測試集上進行評估，結果顯示使用此方法確實在這些 DAOD 任務中取得了顯著的改善，也證明了這個自適應 model 的有效性及在 DAOD 中應用更強大，更快速的 detector 的重要性。



利用 GAN 來進行風格轉換之實例:

(a) (b) (c) (d)

SSDA 和原生的 YOLOV5 結果比較:



(a) Oracle YOLOv5 (b) Source Only YOLOv5 (c) SSDA-YOLO

- **總結**
  在性能方面，YOLOv5 以其超高速度和高精確度聞名。它可以在單個 GPU 上實現即時的 object detection，並在好幾個 object detection benchmarks 上取得了最佳的結果，如 COCO 和 PASCAL VOC，而且與以前的 YOLO 版本相比，YOLOv5 的結構更為簡單，訓練速度更快且檢測精度更高。額透過 UDA 的方式，能夠將 YOLOv5 的泛用性更進一步提高，不管是 ConfMix 又或是 SSDA 都是非常有效且直觀的方法，而實作上也被證明非常有效。

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

- **Confmix – yolov5**

-Advantages

1. 第一個根據 target pseudo detections 的區域信心度來混合 source 和 target domain sample 的混合新技術，並實踐在一階段 dectetor 上
2. 利用了新的漸進式 pseudo labelling 方法，通過在自適應學習過程中逐漸限制信心度指標，實現了對 target 的平滑過渡，從而提高了檢測精度
3. 在適應性性能方面取得了很好的成果，在 Sim10k → Cityscapes 的平均精度（mean Average Precision, mAP）提高了 1.7%，在 KITTI → Cityscapes 提高了 3.7%
4. 很少有研究針對一階段 dectetor（例如 FCOS 或 SSD）進行 UDA 的探討

-Disadvantages

1. 目前的物體檢測模型可以分為兩大類: 一階段和二階段的方法。一階段的方式，如 YOLO 和 FCOS，直接從由 CNN 生成的特徵圖中獲得最終結果，計算效率上非常高，能夠實現接近實時的速度。二階段的方式，如 RCNN，先從第一步 region proposals 生成預測，再利用這些訊息生成分類 label 和 bounding box 坐標，這些模型因其高性能而被廣泛使用，但與一階段的方式相比，它們的速度較慢，但準確性較高，有好有壞。
2. 實測 datasets 較少
3. 會受到 pseudo labelling 品質的影響

- **SSDA – yolov5**

-Advantages

1. 以 YOLOv5 作為 backbone，SSDA-YOLO 可以通過完整的 supervised 學習高效地提取 doamin 新特徵，解決傳統二階段 Faster R-CNN 或 DETR 速度較慢的問題

2. 採用了 knowledge distillation framework，使用 Mean Teacher 來檢測未標記的 target 圖像，過濾預測結果，以迭代方式生成高效的 pseudo label，再實現對 student network 的相對無偏差更新
3. 在圖像級別上縮小教師和學生模型之間的距離，使用優秀的非配對圖像生成方法 CUT 來合成假圖像作為額外的輸入，而非傳統的 CycleGAN
4. 在兩個常見的 doamin 轉換（例如 PascalVOC7→Clipart1k 和 Cityscapes7→Foggy Cityscapes）上實現了優秀的準確度提升，在不同課堂場景下的打呵欠額外驗證中也顯示了其泛化性
5. 以前的方法多使用單個共享網路來擬合 cross-domain 數據，是一種對抗性的過程，大多數利用了雙向操作符梯度反轉層（GRL）等技術，而 knowledge distillation framework 是更好的方式
6. 不使用佔目前主導地位但效率低下過時的 Faster R-CNN，引入了更優越的 YOLOv5 作為主體
7. 只包含簡單的想法架構: Mean Teacher 模型及 knowledge distillation framework 引導的 student network 更新，假的跨 doamin 生成的訓練圖像，用於緩解 image-level domain 差異，配合一致性損失，進一步糾正跨 doman 偏差

-Disadvantages

1. 學生模型的權重更新受到原 domain 中 I s 圖像的的主導
2. 先制條件的 CUT 模型需要依照任務先重新訓練，才能生成出較有效的圖，而且不論是訓練或是預測速度都非常緩慢
3. 結果受到前置步驟 CUT 的生成的圖片品質影響
4. 結果受到 Mean Teacher 過濾的 pseudo label 影響，可能影響到 Student model 的更新
5. 進入第二階段訓練時，pseudo label 產生的品質會大幅影響 domain 轉移效率
6. 訓練時吃資源，因為含有 4 份資料要同時運算(真 target、假 target、家 source、假 source)
7. 實測 datasets 較少

- **Compare**

  兩者各自有好處也有壞處，很難說誰比較實用，ConfMix 跟 SSDA 都利用 YOLOv5 當作主體進行轉移，ConfMix 使用較直觀且簡單的手法進行 UDA，並且效果也非常顯著，而 SSDA 使用較複雜的方式，可能結果有可能條是較，但變數較多，容易影響訓練，且要先使用 GAN 生成假資料，最後訓練時資料量大，很吃硬體資源。兩者都會受到 pseudo labelling 品質的影響，而實際誰好誰壞要用更多 datasets 進行驗證，但 ConfMix 較容易訓練結果也不差確實是個很好的優點，也相對輕量化

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

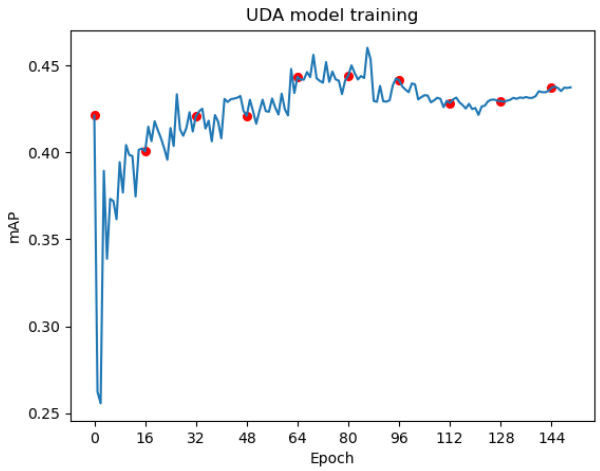
- **ConfMix UDA method: yolov5x/batch size=4 / epochs=151/ optimizer=SGD**
- **mAP@[50:5:95], mAP@50, mAP@75**
  YOLOv5 原生 mAP 列表:

Class	Precision	Recall	mAP50	mAP75	mAP[50:5:95]
All	0.734	0.416	0.459	0.328	0.307
Person	0.767	0.452	0.512	0.305	0.298
Car	0.852	0.617	0.702	0.532	0.507
Truck	0.425	0.258	0.303	0.271	0.246
Bus	0.898	0.43	0.49	0.399	0.379
Rider	0.776	0.467	0.497	0.342	0.304
Motorcycle	0.672	0.402	0.442	0.27	0.251
Bicycle	0.623	0.401	0.392	0.211	0.212
Train	0.859	0.3	0.332	0.296	0.264

Result of check_your_prediction_valid.py:

Class	mAP_small	mAP_medium	mAP_large
All	0.0118	0.1941	0.5394
	mAP50	mAP75	mAP[50:5:95]
	0.4445	0.3083	0.2961

- **mAP curve**



- **Magnitude of improvement in mAP@50 (adapted model - source model):**
  0.4445 / 0.4161 = 約提升 1.068253 倍 (6.82%)

- **如果我將 inference 時的圖片大小調整為 1280 而不是 default 的 640，將有助於提升 mAP50 (因為 training 時使用 640 大小的圖片當作 input，要再現 mAP50 的曲線要使用 640 當作 inference 大小，而最終我是選用 1280 的大小來當作計算排名時使用):**

Result of check_your_prediction_valid.py (image-size = 1280):

Class	mAP_small	mAP_medium	mAP_large
All	0.0774	0.2825	0.4936
	mAP50	mAP75	mAP[50:5:95]
	0.5084	0.3441	0.3277

*** Magnitude of improvement in mAP@50 (adapted model - source model):**
0.5084 / 0.4161 = 約提升 1.221822 倍 (22.18%)

6. Please use either UMAP, PCA, or T-SNE to project the backbone features onto a 2D or 3D space, and include all four features extracted from the following four settings, resulting in a total of 1200 data points. Color-code each data point according to which feature it corresponds to in each setting, and provide a legend to clarify the color scheme. Finally, please briefly discuss any interesting findings from the projection results:

The four settings are:

a. source model (prob.1) - inference on clear-val (300 data)

b. source model (prob.1) - inference on foggy-val (300 data)

c. adapted model (prob.2) - inference on clear-val (300 data)

d. adapted model (prob.2) - inference on foggy-val (300 data)

無作答

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

- **Trained source model-mAP50:**
  YOLOv5 原生 mAP 列表: 0.459

  Result of check_your_prediction_valid.py:

Class	mAP_small	mAP_medium	mAP_large
All	0.0118	0.1941	0.5394
	mAP50	mAP75	mAP[50:5:95]
	0.4445	0.3083	0.2961

- **Random initial or pretrained weights from [COCO, ImageNet] datasets-mAP50:**
  YOLOv5 原生 mAP 列表: 0.483

Result of check_your_prediction_valid.py:

Class	mAP_small	mAP_medium	mAP_large
All	0.0185	0.1888	0.5877
	mAP50	mAP75	mAP[50:5:95]
	0.4743	0.3404	0.3207

- Compare: 與預期不合，direct pretrained weights from COCO 的 yolov5x weight 所獲得的 mAP50 結果比起我利用先在 source domain 訓練完的 weight 再繼續做訓練的高，可能的原因我認為可能有 3 個:

    1. 我所使用的 ConfMix 方法為混合 source domain 及 target domain 的圖像做混合拼接進行訓練，可能對於是否有事先做 source domain 的訓練影響不大，最終兩者都會收斂到類似的 mAP50 位置
    2. 就 ConfMix 的設計架構而言，如果先在 source domain 進行單獨的訓練且收斂到最高點一段時間，可能會導致他過度擬合到 source domain 上而難以轉移到 target domain 上
    3. 我在分兩階段的訓練中第一階段使用的 optimizer 是 AdamW，而直接使用 pretrained weights from COCO 的訓練是使用 SGD，因為時間不夠無法做更多嘗試，但可能就這次的任務來說 SGD optimizer 較合適，而導致出現了偏差

8. Reference:

(1) https://github.com/ultralytics/yolov5

(2) https://github.com/giuliomattolin/ConfMix

(3) https://github.com/hnuzhy/SSDA-YOLO

(4) ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing. Giulio Mattolin, Luca Zanella, Yiming Wang, Elisa Ricci. WACV 2023

(5) SSDA-YOLO: Semi-supervised Domain Adaptive YOLO for Cross-Domain Object Detection. Huayi Zhou, Fei Jiang, Hongtao Lu. Computer Vision and Image Understanding. 2022