Machine Learning and Artificial Intelligence 2020/2021

Domain Adaptive Visual Object Detection

Fereshteh Feizabadi 274475

Mahtab Niknahad 273284

Uditi Ojha 270049

Andrea Zappavigna 278042

Domain Adaptive Visual Object Detection

The goal is to study the visual object detection task and understand what it takes to transfer task knowledge from a source domain with instance-level annotation to an unlabeled target one.

Main issues to be addressed

Even though state-of-art deep object detectors have become extremely precise and faster than ever, all of them record a decrease in performance in domain shifted scenarios.

It is often unrealistic to prepare instance-level annotations for a large number of images in many different domains. There are many obstacles such as lack of copyright-free image sources and cost of annotation.

Project Premise

In our project we start from the results of previous related work ‘Cross-Domain Weakly Supervised Object Detection through Progressive Domain Adaptation’ and we progressively take steps to achieve our goal.

We measure our experimental results against the ones of the reference paper.

Implementation Details

Starting from a fully supervised single-stage object detector, which is pretrained on the source domain, we propose two different domain adaptation strategies by fine-tuning the detector model on different artificially generated samples.

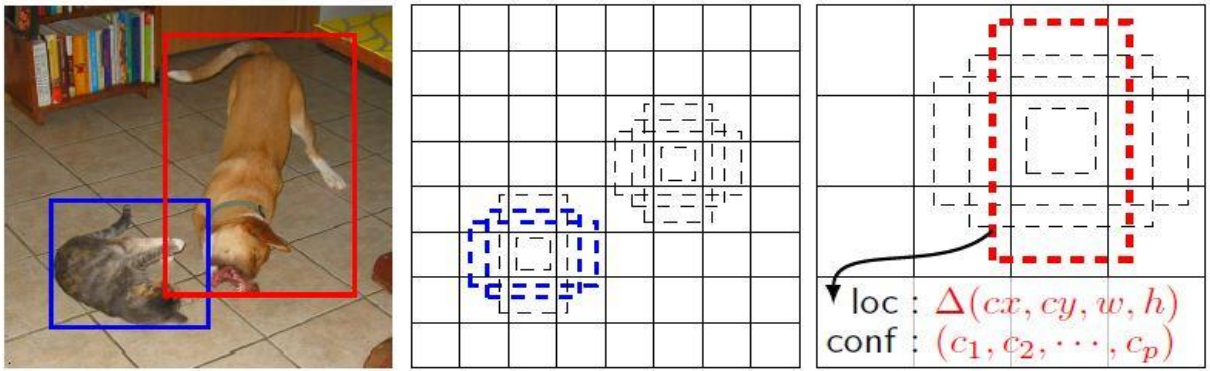
We propose two different pixel-level domain transfer strategies: **CycleGAN** and **AdaIN**, that will allow us to translate the images from the source to the target visual domain.

Dataset

Source domain: The natural images in PASCAL VOC, Overall including 20 classes. In particular, we use the VOC2007 and VOOC2012 trainval splits for a grand total of 16551 images.

Target domain: the Clipart1k dataset is used. It contains the same 20 classes, with 1000 images and 3165 instances.

Single Shot Multibox Detector (SSD)



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

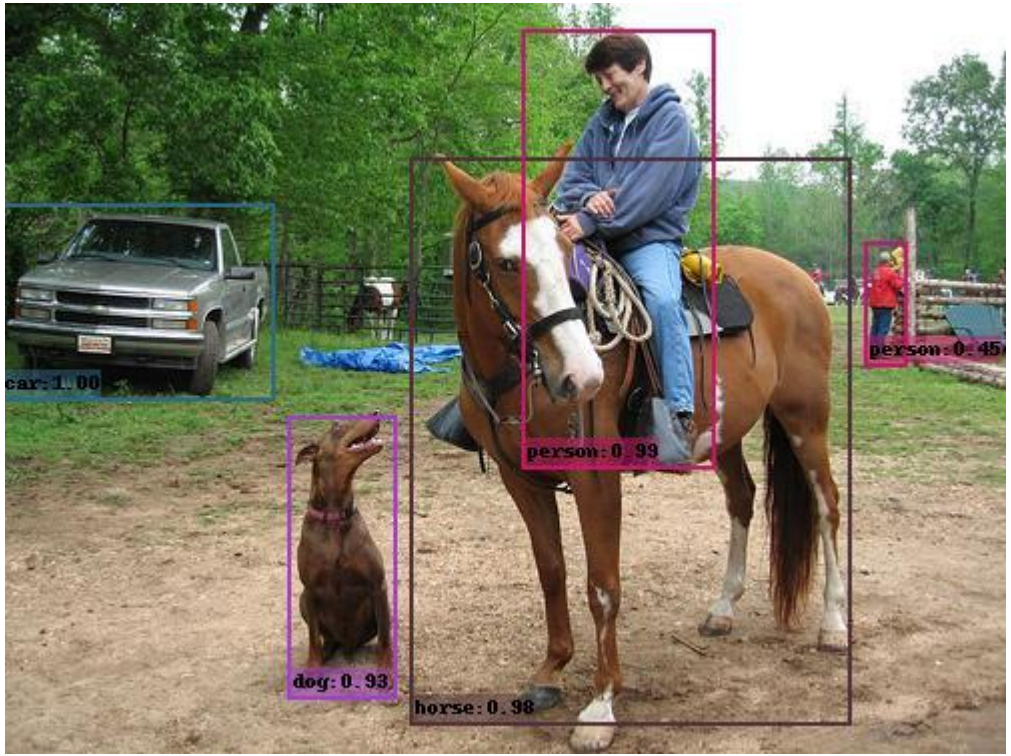
SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates for the presence of each object category in each default box.

SSD Framework

- SSD only needs an input image and **ground truth boxes for each object** during training.
- It evaluates a **small set of default boxes** of different aspect ratios at each location in several feature maps with different scales.
- For each default box, it **predicts both the shape offsets and the confidences** for all object categories.
- At training time, it **matches these default boxes to the ground truth boxes**.
- The model loss is a weighted sum between **localization loss** (e.g. Smooth L1) and **confidence loss** (e.g. Softmax).

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

SSD Outputs



Example SSD output
(vgg_ssd300_voc0712)

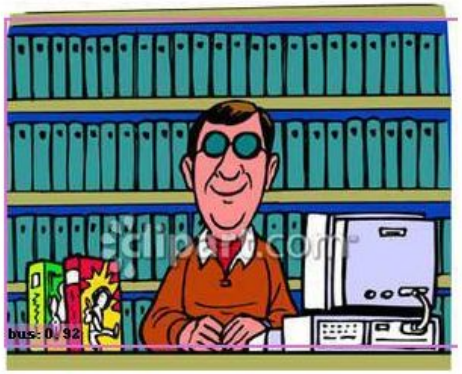
SSD Baseline

Training: We train SSD300 model in the source domain with instance-level annotations starting with 40k iterations at 1e-3 learning rate, we then continue with lr 1e-4 and 1e-5 for 10k iterations each (total 60k).

An 0.5 IoU threshold, also known as Jaccard Index, for NMS and 0.1 confidence threshold for discarding low confidence detections were employed.

Testing: We test our SSD trained model in the target domain, which is a different image domain with respect to the source domain that our model was trained on.

SSD Baseline Results



The baseline model fails to achieve stable results: in the first image only 1 out of 3 cars is detected, it outrights fails to recognize the dog in the middle image and it mistakenly detects a bus in the last image.

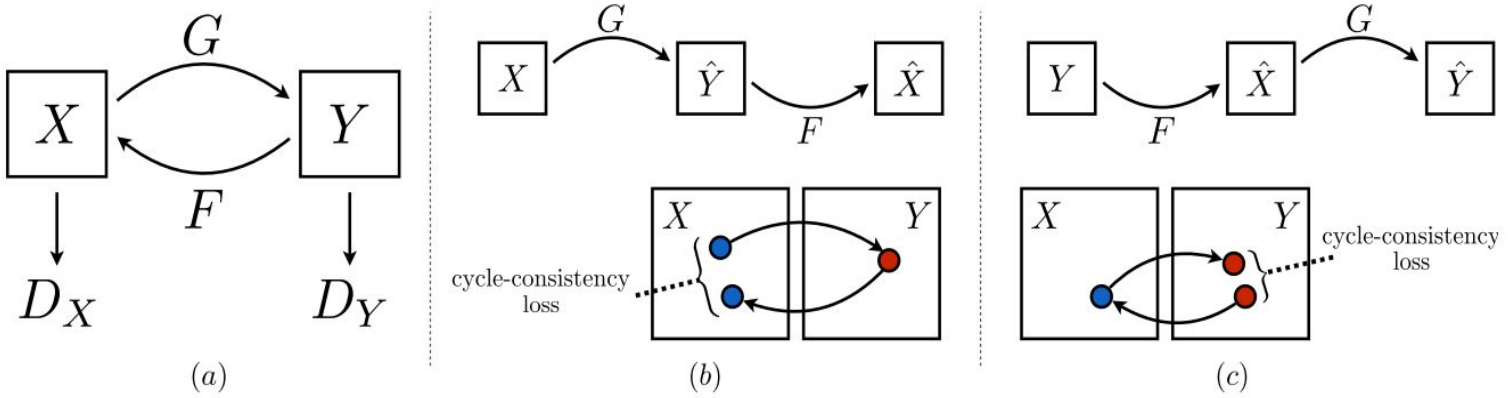
CycleGan: DT1

The goal of the domain transfer is to generate images that look like the target domain from the source domain.

To achieve this goal, we are using an unpaired image-to-image translation method called CycleGAN. The goal is to learn the mapping functions between two image domains X and Y with unpaired examples.

In practice, a mapping $G : X \rightarrow Y$ and an inverse mapping $F : Y \rightarrow X$ are jointly learned using CNN.

CycleGan: DT1



CycleGan: DT1

Adversarial Loss:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) =& \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],\end{aligned} \tag{1}$$

Cycle Consistency Loss:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) =& \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned} \tag{2}$$

CycleGan: DT1

Training: We trained CycleGAN with a learning rate of 1e-5 for the first 10 epochs and a linear decaying rate to 0 over the next 10 epochs.

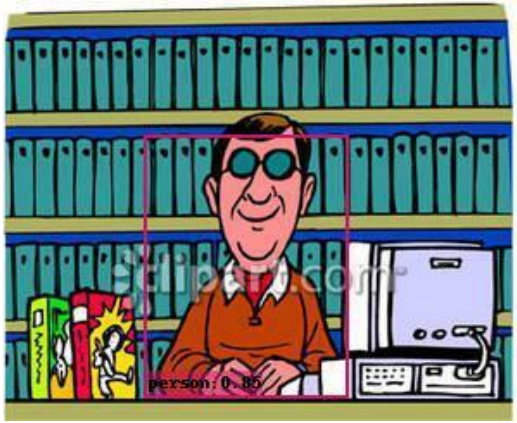
Testing: Once the mapping functions are trained, we convert images in the source domain that are used in the pre-training and obtain domain transferred images that accompany instance-level annotations. Using these images, we fine tune our baseline.

CycleGAN Images






DT1 Results



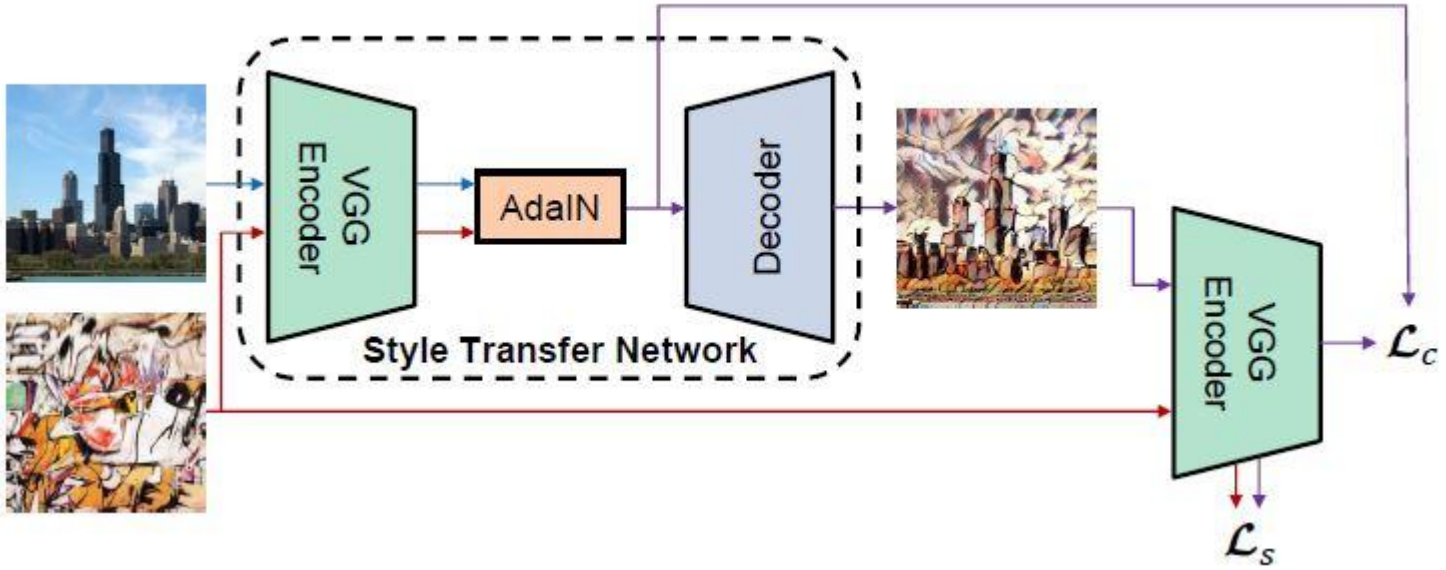
Pictures show some sample output of DT1 model. In the left picture, one of the cars is detected and the other part of the image is wrongly classified.

Arbitrary Style Transfer

Adaptive Instance Normalization is a method that allows to perform style transfer in real-time, at the heart of the model is a novel layer that aligns the content of the source images with the style of the target images.

In short, this step allows us to perform style transfer in the image feature space by transferring feature statistics, specifically the channel-wise mean and variance.

AdaIN Architecture



AdaIN Layer

$$AdaIN(x, y) = \sigma(y)\frac{x - \mu(x)}{\sigma(x)} + \mu(y)$$

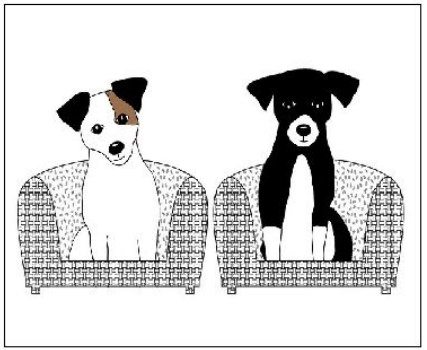
This step effectively combines the content input *x* and the style input *y* by transferring features statistics, specifically the channel-wise mean and variance, so that the values of *x* and *y* match.

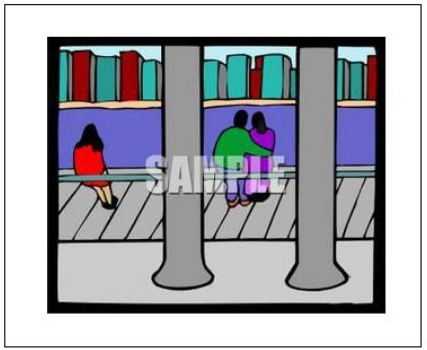
AdaIN: DT2

In order to use AdaIN as a domain adaptation strategy, we apply the translation online so that we can exploit the advantages of the high variability.

In practice, we used the translation model to stylize source images right before sending them to the detector. This step is conditionally executed with probability p each time a new image batch is loaded, or in other words, for each iteration.

Style Transferred Images Through AdaIN

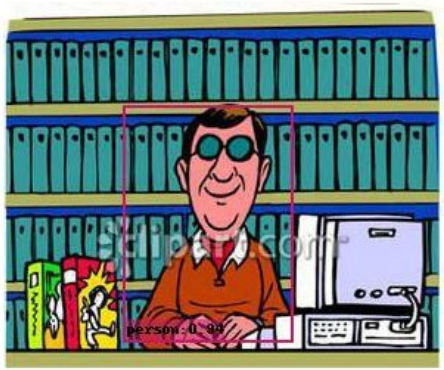



SSD Model: fine-tuning with AdaIN

In this configuration setting, we fine-tune the baseline model using the images transferred through the AdaIN model for 10k iterations with learning rate 1e-5, probability 0.5.

We test our model on the target domain and see an increase in performance compared with the baseline and the previous DT1. This time our results are also in line with the reference paper domain transfer step.

DT2 Results



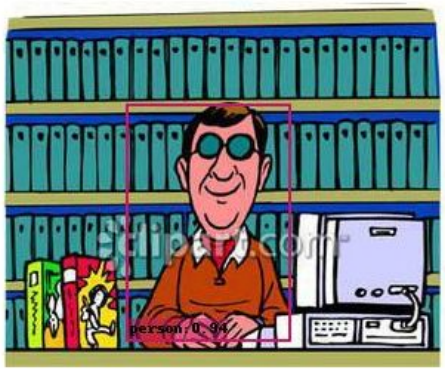
In the first image we see DT2 is able to detect two cars correctly this time, in the second and in the third image it detects like DT1, but with greater confidence score.

Ideal Case

In this experimental setting , we has access to the training split of the target dataset (500 images). We simply fine-tune the baseline model of SSD for 10K iterations with the learning rate of 1e-5 using these images with instance-level annotations.

As always, we test our model on the testing split of the target domain dataset.

Ideal Case Results



Outputs for our ideal case in the target domain. We see here in all the images the detection is correct and the confidence scores are high.

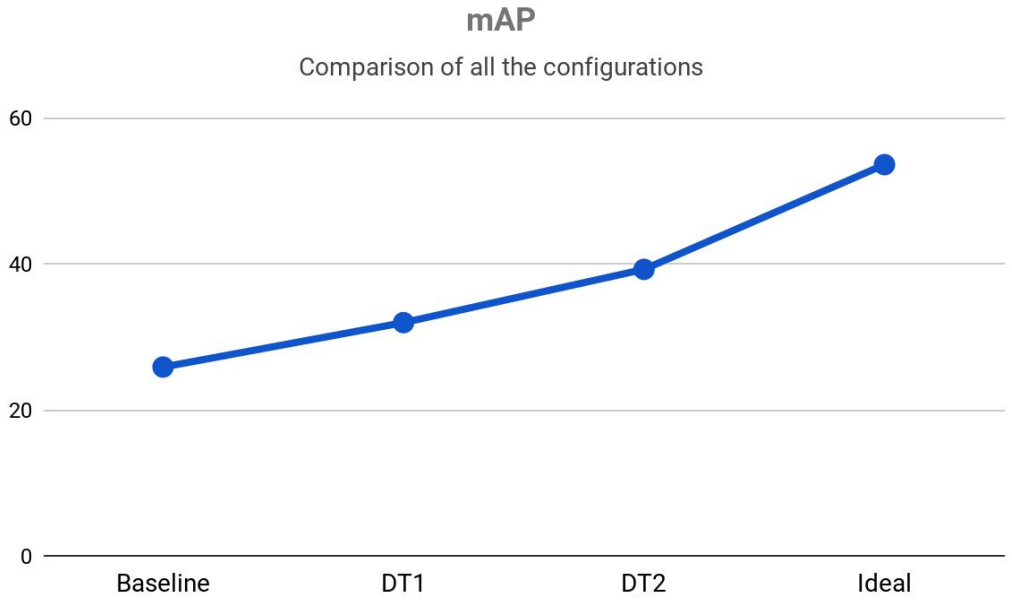
Comparison of all the methods

Baseline: The result is **25.86**, inline with the reference paper result of **26.8**

DT1: The result is **31.95** which gives an improvement of **6 percentage** points from the baseline

DT2: The result is **39.25** which gives an improvement of **13 percentage** w.r.t to the baseline and **7 percentage** w.r.t. DT1

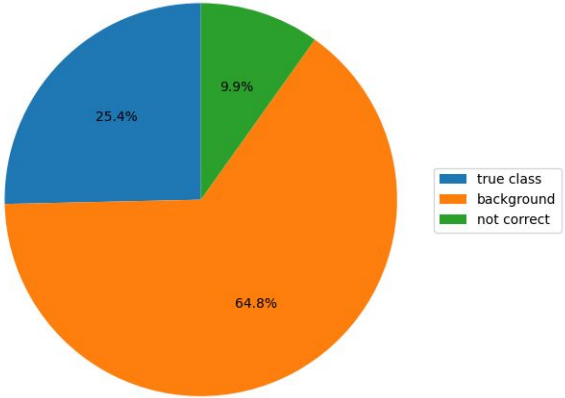
Ideal: The result we obtain in ideal case were **53.60,** slightly below the reference paper result of **55.4**

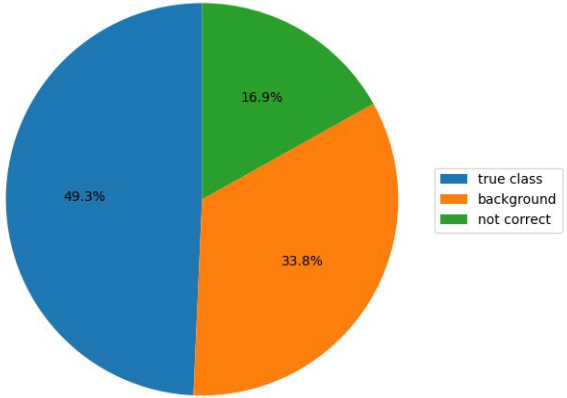


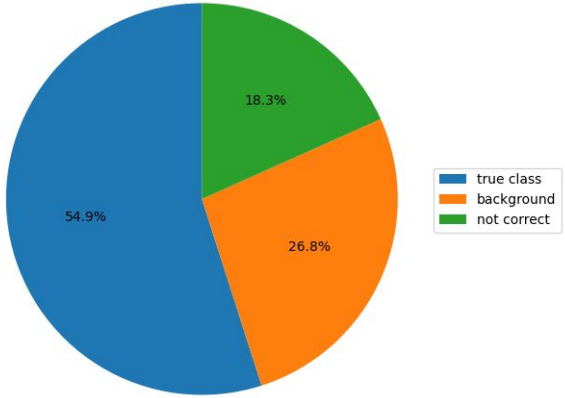
Experimental Results

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mAP**
Reference																					
Baseline	19.8	49.5	20.1	23.0	11.3	38.6	34.2	2.5	39.1	21.6	27.3	10.8	32.5	54.1	45.3	31.2	19.0	19.5	19.1	17.9	26.8
DT	23.3	60.1	24.9	41.5	26.4	53.0	44.0	4.1	45.3	51.5	39.5	11.6	40.4	62.2	61.1	37.1	20.9	39.6	38.4	36.0	38.0
Ideal	50.5	60.3	40.1	55.9	34.8	79.7	61.9	13.5	56.2	76.1	57.7	36.8	63.5	92.3	76.2	49.8	40.2	28.1	60.3	74.4	55.4
Proposed																					
Baseline	19.2	48.7	20.5	18.0	14.6	34.4	30.5	11.6	44.3	10.9	27.5	13.0	25.4	54.4	39.9	26.2	20.0	25.3	19.9	22.9	25.9
DT1	23.5	56.8	22.9	20.8	18.0	35.7	38.6	7.0	46.5	42.6	29.9	9.3	29.8	67.8	54.2	27.9	18.8	37.5	29.5	32.0	32.0
DT2	29.7	63.8	22.8	24.8	20.4	46.6	44.5	5.0	49.5	54.8	40.5	21.0	34.5	74.6	62.4	33.1	18.4	45.8	38.5	54.2	39.3
Ideal	49.6	69.6	40.7	50.1	32.5	66.6	57.6	16.3	53.4	69.2	61.8	27.2	55.6	77.6	73.8	40.5	43.2	55.8	64.4	66.7	53.6

Result On Low-Confident Detections







Comparing baseline with DT1 and DT2, we observed that fine tuning FSD on images obtained by DT1 and DT2 , improves performance, especially in less-confident detections (confidence greater than or equal to 0.1)

Conclusion

- We are able to show that it is indeed possible to perform visual object detection in a unlabeled target domain.
- We were able to achieve improvements by using two different domain adaptation methods, both offline (DT1) and online (DT2)
- Experimental results demonstrated that our domain adaptation methods show better performance when compared with the baseline.

Future Improvements

- Train the baseline model for 80k iterations, to match or improve the reference paper score.
- Improve on the existing CycleGAN model, to reach the reference paper DT score.
- Experiment with a different style-transfer strategy: choose with probability p for each image instead of the existing batch-level decision.
- Experiment with more advanced style-transfer methods (structure-emphasized multimodal style transfer SEMST)
- Experiment with other domain adaptation methods other then pixel level domain adaptation.

Thank you
For your attention