One-Shot Unsupervised Cross-Domain Detection Supplementary Material

Antonio D'Innocente[1,3], Francesco Cappio Borlino[2], Silvia Bucci[2,3], Barbara Caputo[2,3], and Tatiana Tommasi[2,3]

[1] Sapienza University of Rome, Rome, Italy dinnocente@diag.uniroma1.it
[2] Politecnico di Torino, Turin, Italy
{francesco.cappio,silvia.bucci,barbara.caputo,tatiana.tommasi}@polito.it
[3] Italian Institute of Technology, Turin, Italy

1 Full target results

For completeness with respect to the main submission, we extend the original results on VOC → AMD in Table 1 and Cityscapes → FoggyCityscapes in Table 2, reporting the full target results of DivMatch [5] and SW [7]. Since we are granting to those methods the access to a much larger set of data with respect to OSHOT, the reported results can be only considered as an empirical upper-bound. We highlight that OSHOT outperforms SW also in this Full Target case, while remaining lower than DivMatch.

2 Pseudo-labeling

The role of rotation recognition on the whole image for classification across domains was extensively discussed in [9]. The naïve application of this approach would not lead to any advantage in detection: we deal with scenes and the standard position of sky and ground would steer the recognition, not providing useful information. OSHOT uses rotation recognition by focusing explicitly on objects. We consider cropped portions of the image features map to take advantage of the adaptive power of rotation recognition on local information. Indeed, as discussed in [7], local domain alignment is crucial for cross-domain detection.

Although [2,3,4] indicate that pseudo-labeling is a viable option for detection across domains, this approach has some well-known issues, *e.g.* misclassified outputs with high confidence need to be discarded [4]. OSHOT is based on a new form of cross-task pseudo-labeling that avoids those issues: the pseudo-labels produced by the source models are used for the self-supervised rotation classifier. In this way we keep the advantage of model initialization through pseudo-labeling with a reduced risk of error propagation. To provide a practical example: using standard pseudo-labeling (SPS) on VOC → AMD we get the results in the top part of Table 3, where the negative effect of SPS is clearly visible.

Table 1. mAP results for VOC → AMD

(a) VOC → Clipart

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
One-Shot *Target*																					
Source Only																					
FRCNN	18.5	43.3	20.4	13.3	21.0	47.8	29.0	16.9	28.8	12.5	19.5	17.1	23.8	40.6	34.9	34.7	9.1	18.3	40.2	38.0	26.4
OSHOT ($\gamma = 0$)	23.1	55.3	22.7	21.4	26.8	53.3	28.9	4.6	31.4	9.2	27.8	9.6	30.9	47.0	38.2	35.2	11.1	20.4	36.0	33.6	28.3
OSHOT ($\gamma = 10$)	25.4	61.6	23.8	21.1	31.3	55.1	31.6	5.3	34.0	10.1	28.8	7.3	33.1	59.9	44.2	38.8	15.9	19.1	39.5	33.9	31.0
OSHOT ($\gamma = 30$)	25.4	56.0	24.7	25.3	36.7	58.0	34.4	5.9	34.9	10.3	29.2	11.8	46.9	70.9	52.9	41.5	21.1	21.0	38.5	31.8	**33.9**
Ten-Shot *Target*																					
DivMatch [5]	19.5	57.2	17.0	23.8	14.4	25.4	29.4	2.7	35.0	8.4	22.9	14.2	30.0	55.6	50.8	30.2	1.9	12.3	37.8	37.2	26.3
SW [7]	21.5	39.9	21.7	20.5	32.7	34.1	25.1	8.5	33.2	10.9	15.2	3.4	32.2	56.9	46.5	35.4	14.7	15.2	29.2	32.0	26.4
Full *Target*																					
DivMatch [5]	26.6	66.7	26.7	33.9	42.4	35.4	40.7	12.0	45.9	51.6	23.5	15.5	49.1	71.6	64.0	45.3	20.4	36.6	45.3	41.3	39.7
SW [7]	15.5	37.1	22.9	22.1	32.6	44.6	28.1	10.1	34.6	9.8	19.2	11.5	36.5	56.9	57.1	37.4	12.3	9.6	33.2	34.2	28.3

(b) VOC → Comic

Method	bike	bird	car	cat	dog	person	mAP
One-Shot *Target*							
FRCNN	25.2	10.0	21.1	14.1	11.0	27.1	18.1
OSHOT ($\gamma = 0$)	26.9	11.6	22.7	9.1	14.2	28.3	18.8
OSHOT ($\gamma = 10$)	35.5	11.7	25.1	9.1	15.8	34.5	22.0
OSHOT ($\gamma = 30$)	35.2	14.4	30.0	14.8	20.0	46.7	**26.9**
Ten-Shot *Target*							
DivMatch [5]	27.1	12.3	26.2	11.5	13.8	34.0	20.8
SW [7]	21.2	14.8	18.7	12.4	14.9	43.9	21.0
Full *Target*							
DivMatch [5]	59.3	20.5	35.5	21.0	30.6	54.6	36.9
SW [7]	39.9	16.2	23.0	11.5	13.5	44.5	24.8

(c) VOC → Watercolor

Method	bike	bird	car	cat	dog	person	mAP
One-Shot *Target*							
FRCNN	62.5	39.7	43.4	31.9	26.7	52.4	42.8
OSHOT ($\gamma = 0$)	70.2	46.7	45.5	31.2	27.2	55.7	46.1
OSHOT ($\gamma = 10$)	70.2	46.7	48.1	30.9	32.3	59.9	48.0
OSHOT ($\gamma = 30$)	77.1	44.7	52.4	37.3	37.0	63.3	**52.0**
Ten-Shot *Target*							
DivMatch [5]	64.6	44.1	44.6	34.1	24.9	60.0	45.4
SW [7]	66.3	41.1	41.1	30.5	20.5	52.3	42.0
Full *Target*							
DivMatch [5]	86.6	50.5	50.9	36.2	39.5	64.5	54.7
SW [7]	65.7	45.1	43.1	32.4	30.5	56.2	45.5

Table 2. mAP results for Cityscapes → FoggyCityscapes

Method	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
One-Shot *Target*									
Source Only									
FRCNN	30.4	36.3	41.4	18.5	32.8	9.1	20.3	25.9	26.8
OSHOT ($\gamma = 0$)	31.8	42.0	42.6	20.1	31.6	10.6	24.8	30.7	29.3
OSHOT ($\gamma = 10$)	31.9	41.9	43.0	19.7	38.0	10.4	25.5	30.2	30.1
OSHOT ($\gamma = 30$)	32.1	46.1	43.1	20.4	39.8	15.9	27.1	32.4	**31.9**
Ten-Shot *Target*									
DivMatch [5]	27.6	38.1	42.9	17.1	27.6	14.3	14.6	32.8	26.9
SW [7]	25.5	30.8	40.4	21.1	26.1	34.5	6.1	13.4	24.7
Full *Target*									
DivMatch [5]	32.3	43.5	47.6	23.9	38.0	23.1	27.6	37.2	34.2
SW [7]	31.3	32.1	47.4	19.6	28.8	41.0	9.8	20.1	28.8

3 Self-Supervision

For the auxiliary self-supervised task we set the loss weight $\lambda = 0.05$ in the pretraining stage and increase it to $\lambda = 0.2$ for adaptation. This choice has the same effect as increasing the learning rate during adaptation and helps to speed up the inference stage. Using a lower weight for λ would require more finetuning iterations, as shown in the middle part of Table 3.

Besides rotation recognition it is also possible to use other self-supervised tasks for the OSHOT algorithm. In the last two rows of Table 3, we evaluate also the suitability of self-supervised jigsaw puzzle [6]. The obtained results confirm the effectiveness of our method.

Table 3. Analysis on pseudo-labeling and self-supervised tasks

Method	Clipart	Comic	Watercolor	Avg
FRCNN	26.4	18.1	42.8	29.1
OSHOT ($\gamma = 30$)	33.9	26.9	52.0	37.6
Standard Pseudo-labeling				
OSHOT SPS ($\gamma = 10$)	17.4	14.0	21.7	17.7
OSHOT SPS ($\gamma = 30$)	6.7	5.3	12.1	6.0
Self-Supervised Weight and Iterations				
OSHOT ($\gamma = 30$, $\lambda = 0.05$)	33.8	23.4	50.7	36.0
OSHOT ($\gamma = 70$, $\lambda = 0.05$)	34.1	24.5	51.2	36.6
OSHOT ($\gamma = 30$, $\lambda = 0.2$)	33.9	26.9	52.0	37.6
Self-Supervised Jigsaw Task				
OSHOT Jigsaw ($\gamma = 0$)	29.0	18.9	42.8	30.2
OSHOT Jigsaw ($\gamma = 30$)	27.1	16.0	40.5	27.9

4 Full Ablation Results

Detection error analysis We complete here the detection error analysis that was only partially included in the main paper for space reasons. Specifically we consider all the three domain shift cases of VOC $\rightarrow$ AMD together with Cityscapes $\rightarrow$ Foggy Cityscapes, KITTI $\rightarrow$ Cityscapes and KITTI $\rightarrow$ Cityscapes. As reported in the main paper, for the first benchmark VOC $\rightarrow$ Clipart we follow [1,5] considering the top $1k$ most confident detections and identifying three error types: correct (IoUgt $\geqslant 0.5$), mislocalized ($0.3 \leqslant$ IoUgt < 0.5) and background (IoUgt < 0.3). For VOC $\rightarrow$ Comic and VOC $\rightarrow$ Watercolor we consider $2k$ most confident predictions, maintaining the same ratio of the first case given that the number of target samples is twice that of Clipart. A similar reasoning, that also takes care of the class cardinality, was applied to choose 6k most confident predictions for Foggy Cityscapes $\rightarrow$ Cityscapes, 1.5k for KITTI $\rightarrow$ Cityscapes and 20k for Cityscapes $\rightarrow$ KITTI. From Figure 1 we can state that for both Clipart and Watercolor the advantage of adding the self-supervised task at training time is limited ($\gamma = 0$), while the gain becomes evident when the number of adaptive iterations grows ($\gamma = 30$). For Comic the improvement in performance appears already in the pretraining phase and further increases with adaptation. Overall the false positive errors decrease, while the ratio between the mislocalization error and correct localizations either decreases (Clipar, Comic) or remains stable (Watercolor). A similar behaviour can be observed on the urban scenes, both when testing on Foggy Cityscapes and Cityscapes, as shown in the first two rows of Figure 2. For the last case of testing on KITTI, the results remain almost stable, confirming the same trend observed on the overall mAP performance discussed in the main paper. A neglegible drop of 0.7% correct predictions appear when applying the adaptation phase for $\gamma = 30$.

Self-supervised iterations We report results of OSHOT at different number of self-supervised iterations in Figure 3. We observe positive correlations between

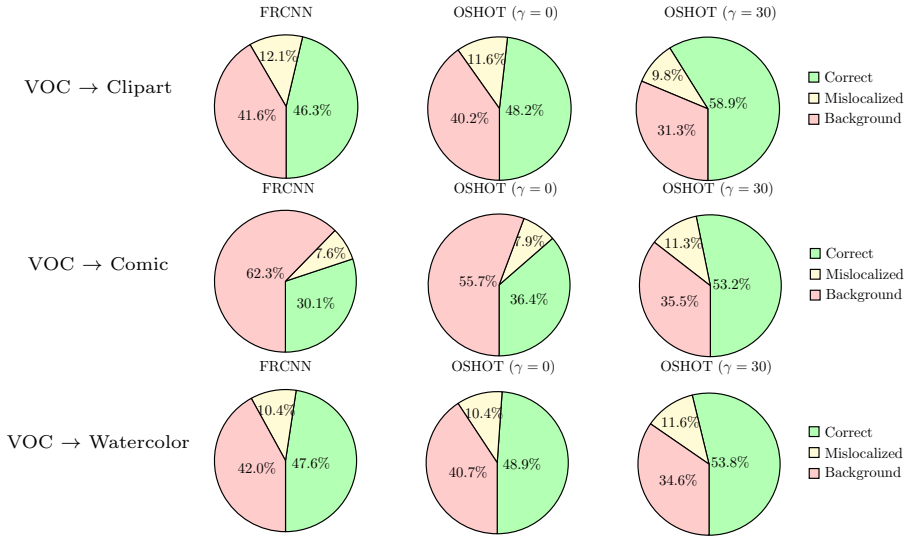

Fig. 1. Detection error analysis on the three cases of VOC → AMD

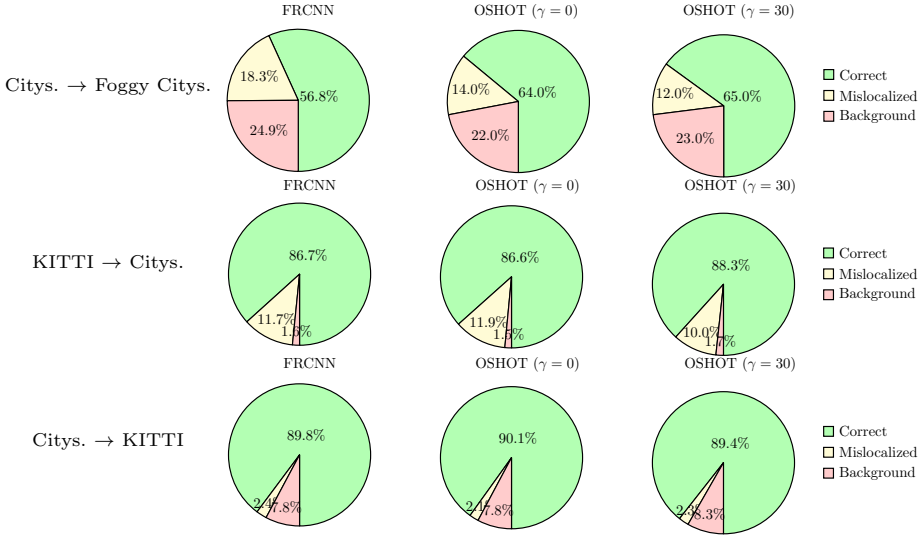

Fig. 2. Detection error analysis on the three cases of urban scenes

the number of self-supervised iterations and the final mAP on all targets except KITTI, for which the final results are minimally affected by our adaptation procedure (as well as by any other adaptive method used as reference - see Table 5 of the main paper). The first 10 iterations show the most significant mAP change, while it gets to a stable plateau for further iterations.

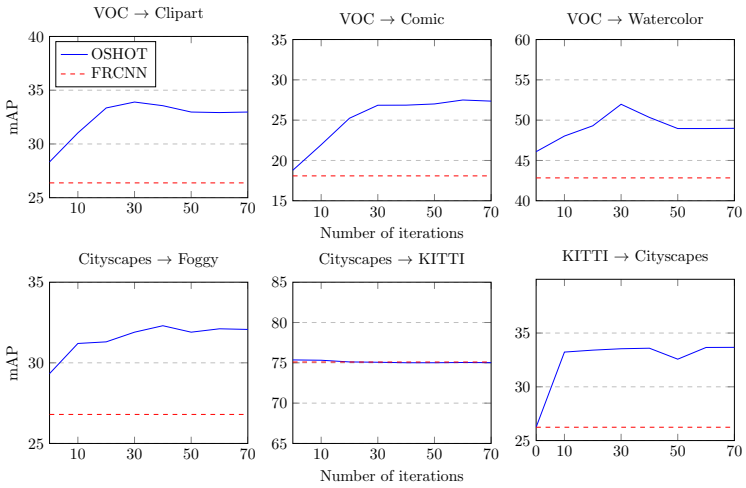

Fig. 3. OSHOT at different number of iterations for all testbeds

5 Qualitative Analysis

Image vs Box rotation To validate our choice of considering box rotation over image rotation we set up a dedicated experiment. We ran the pretraining stage of OSHOT on VOC by using either $G_r(image)$ or $G_r(boxcrop)$. Then we tested the rotation classifier on whole images from the Clipart domain. In Figure 4 we show the results obtained with Grad-CAM [8] for the two cases, with heatmap indicating the most relevant regions responsible for recognizing the correct orientation. The Grad-CAM maps refer to the last output of the backbone feature extractor. We can see that, when the rotation classifier is trained on whole *images* it learns to focus on the background (*e.g.* the sky and the ground) in order to solve the task. On the contrary, when the *boxcrop* operation is implemented to train the rotation classifier only on the relevant objects, it learns to look at objects' features even when it faces an entire image.

Detection results of OSHOT: baselines and self-supervised iterations Figure 5 shows some examples of detections of OSHOT on images extracted from all the datasets considered in our work. We present as reference also the ground truth results as well as the predictions produced by DivMatch [5] and SW [7] that appear less precise than OSHOT.

6 OSHOT pseudocode

The pseudocode for the adaptive phase of OSHOT is presented in Algorithm 1. Here, G_f and G_d indicate the backbone feature extractor and detector, respectively parametrized by θ_f and θ_d. FC is the fully connected layer of the rotation

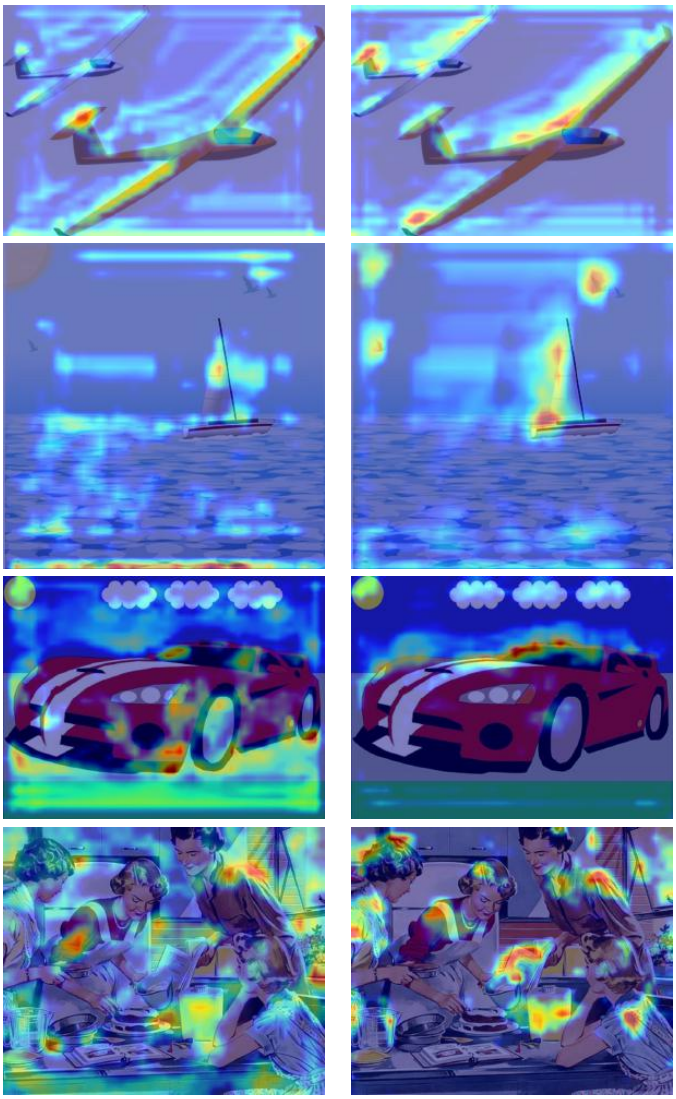

Fig. 4. Visualization of the most relevant image regions produced by Grad-CAM when classifying the correct rotation with $G_r(image)$ and $G_r(boxcrop)$

classifier, parametrized by θ_r, and R is the rotator operator $R(x, \alpha)$ where α, which indicates one random rotation to apply on x, is dropped for simplicity. The $pseudoboxcrop(\cdot|b)$ operator applies cropping and ROI pooling on the input feature map based on the corresponding location of pseudo-boxes b.

Fig. 5. Qualitative visualization of detections with DivMatch, SW and OSHOT on Comic (first row), Foggy Cityscapes (second row), Watercolor (third row), Social Bikes (fourth and fifth rows) and KITTI (sixth row). Numbers associated with bounding boxes indicate the detector's confidence

Algorithm 1: Adaptive phase of OSHOT

Data: G_f, G_d, FC, parameters θ_f, θ_r, θ_d, rotator R, target image x^t
$\theta_f^* \leftarrow \theta_f$
$\theta_r^* \leftarrow \theta_r$
while *still* γ *iterations* **do**
  $b^t, c^t \leftarrow G_d(G_f(x^t|\theta_f^*)|\theta_d)$
  $x_r^t \leftarrow R(x^t)$
  $b_r^t \leftarrow R(b^t)$
  minimize self-supervised loss $\mathcal{L}_r(FC_{\theta_r^*}(pseudoboxcrop(G_f(x_r^t|\theta_f^*)|b_r^t))$
  update θ_f^*, θ_r^*
end
predict label of test sample using $\mathcal{G}_f(\cdot|\theta_f^*), \mathcal{G}_d$

References

1. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV (2012)
2. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018)
3. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: ICCV (2019)
4. Kim, S., Choi, J., Kim, T., Kim, C.: Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In: ICCV (2019)
5. Kim, T., Jeong, M., Kim, S., Choi, S., Kim, C.: Diversify and match: A domain adaptive representation learning paradigm for object detection. In: CVPR (2019)
6. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
7. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: CVPR (2019)
8. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
9. Xu, J., Xiao, L., López, A.M.: Self-supervised domain adaptation for computer vision tasks. ArXiv **abs/1907.10915** (2019)