One-Shot Unsupervised Cross-Domain Detection

One-Shot Unsupervised Cross-Domain Detection

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

Abstract. Despite impressive progress in object detection over the last years, it is still an open challenge to reliably detect objects across visual domains. All current approaches access a sizable amount of target data at training time. This is a heavy assumption, as often it is not possible to anticipate the domain where a detector will be used, nor to access it in advance for data acquisition. Consider for instance the task of monitoring image feeds from social media: as every image is uploaded by a different user it belongs to a different target domain that is impossible to foresee during training. Our work addresses this setting, presenting an object detection algorithm able to perform unsupervised adaptation across domains by using only one target sample, seen at test time. We introduce a multi-task architecture that one-shot adapts to any incoming sample by iteratively solving a self-supervised task on it. We further enhance this auxiliary adaptation with cross-task pseudo-labeling. A thorough benchmark analysis against the most recent cross-domain detection methods and a detailed ablation study show the advantage of our approach.

Keywords: Object detection · Cross-domain analysis · Self-supervision

1 Introduction

Social media feed us every day with an unprecedented amount of visual data. Images are uploaded by various actors, from corporations to political parties, institutions, entrepreneurs and private citizens, with roughly $10^2 M$ unique images shared everyday on Twitter, Facebook and Instagram. For the sake of freedom of expression, control over their content is limited, and their vast majority is uploaded without any textual description of their content. Their sheer magnitude makes it imperative to use algorithms to monitor and make sense of them, finding the right balance between protecting the privacy of citizens and their right of expression, and tracking fake news (often associated with malicious intentions) while fighting illegal and hate content. This in most cases boils down to the ability to automatically associate as many tags as possible to images, which in turns means determining which objects are present in a scene.

Object detection has been largely investigated since the infancy of computer vision [47,11] and continues to attract a large attention in the current deep

learning era [19,10,52,30]. Most of the algorithms assume that training and test data come from the same visual domain [19,18,40]. Recently, some authors have started to investigate the more challenging yet realistic scenario where the detector is trained on data from a visual source domain, and deployed at test time in a different target domain [32,33,44,46]. This setting is usually referred to as *cross-domain detection* and heavily relies on concepts and results from the domain adaptation literature [32,14,20]. Specifically, it inherits the standard tansductive logic, according to which unsupervised target data is available at training time together with annotated source data, and can be used to adapt across domains. This approach is not suitable, neither effective, for monitoring social media feeds. Consider for instance the scenario depicted in Figure 1, where there is an incoming stream of images from various social media and the detector is asked to look for instances of the class bicycle. The images come continuously, but they are produced by different users that share them on different social platforms. Hence, even though they might contain the same object, each of them has been acquired by a different person, in a different context, under different viewpoints and illuminations. In other words, *each image comes from a different visual domain, distinct from the visual domain where the detector has been trained*. This poses two key challenges to current cross-domain detectors: (1) to adapt to the target data, these algorithms need first to gather feeds, and only after enough target data has been collected they can learn to adapt and start performing on the incoming images; (2) even if the algorithms have learned to adapt on target images from the feed up to time t, there is no guarantee that the images that will arrive from time $t + 1$ will come from the same target domain.

This is the scenario we address. We focus on cross-domain detection when only one target sample is available for adaptation, without any form of supervision. We propose an object detection method able to adapt from one target image, hence suitable for the social media scenario described above. Specifically, we build a multi-task deep architecture that adapts across domains by leveraging over a pretext task. This auxiliary knowledge is further guided by a cross-task pseudo-labeling that injects the locality specific of object detection into self-supervised learning. The result is an architecture able to perform unsupervised adaptive object detection from a single image. Extensive experiments show the power of our method compared to previous state-of-the-art approaches. To summarize, the contributions of our paper are as follows:
(1) we introduce the One-Shot Unsupervised Cross-Domain Detection setting, a cross-domain detection scenario where the target domain changes from sample to sample, hence adaptation can be learned only from one image. This scenario is especially relevant for monitoring social media image feeds. We are not aware of previous works addressing it.
(2) We propose OSHOT, the first cross-domain object detector able to perform one-shot unsupervised adaptation. Our approach leverages over self-supervised one-shot learning guided by a cross-task pseudo-labeling procedure, embedded into a multi-task architecture. A thorough ablation study showcases the importance of each component.

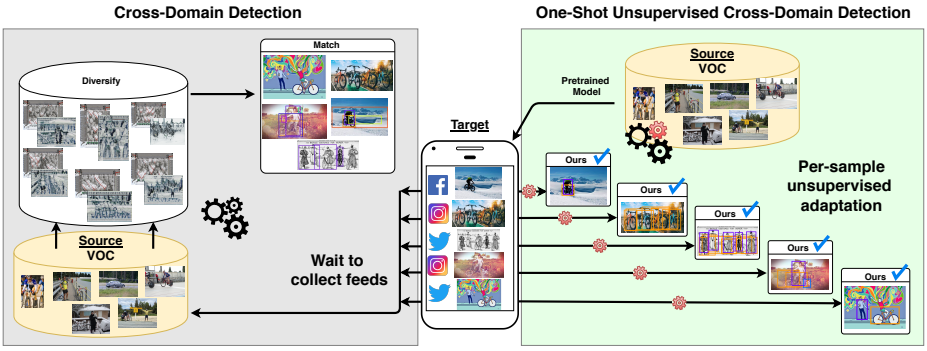


Fig. 1. Each social media image comes from a different domain. Existing Cross-Domain Detection algorithms (*e.g.* [28] in the left gray box) struggle to adapt in this setting. OSHOT (right) is able to adapt across domains from one single target image, thanks to the combined use of self-supervision and pseudo-labeling

(3) We present a new experimental setup for studying one-shot unsupervised cross-domain adaptation, designed on three existing databases plus a new test set collected from social media feed. We compare against recent algorithms in cross-domain adaptive detection [42,28] and one-shot unsupervised learning [8], achieving the state-of-the-art.

We make the code of our project available at `https://github.com/VeloDC/oshot_detection`.

2 Related Work

Object Detection Many successful object detection approaches have been developed during the past several years, starting from the original sliding window methods based on handcrafted features, till the most recent deep-learning empowered solutions. Modern detectors can be divided into *one-stage* and *two-stage* techniques. In the former, classification and bounding box prediction is performed on the convolution feature map either solving a regression problem on grid cells [39], or exploiting anchor boxes at different scales and aspect ratios [31]. In the latter, an initial stage deals with the region proposal process and is followed by a refinement stage that adjusts the coarse region localization and classifies the box content. Existing variants of this strategy differ mainly in the region proposal algorithm [19,18,40]. Regardless of the specific implementation, the detector robustness across visual domains remains a major issue.

Cross-Domain Detection When training and test data are drawn from two different distributions a model learned on the first is doomed to fail on the second. Unsupervised domain adaptation methods attempt to close the domain gap between the annotated source on which learning is performed, and the target

samples on which the model is deployed. Most of the literature has focused on object classification with solutions based on feature alignment [32,33,44,2] or adversarial approaches [15,46]. GAN-based methods allow to directly update the visual style of the annotated source data and reduce the domain shift directly at pixel level [41,23]. Only in the last two years adaptive detection methods have been developed considering three main components: (i) including multiple and increasingly more accurate feature alignment modules at different internal stages, (ii) adding a preliminary pixel-level adaptation and (iii) pseudo-labeling. The last one is also known as *self-training* and consists in using the output of the source model detector as coarse annotation on the target. The importance of considering both global and local domain adaptation, together with a consistency regularizer to bridge the two, was first highlighted in [7]. The Strong-Weak (SW) method of [42] improves over the previous one pointing out the need of a better balanced alignment with strong global and weak local adaptation. It was also further extended by [49], where the adaptive steps are multiplied at different depth in the network. By generating new source images that look like those of the target, the Domain-Transfer (DT, [25]) method was the first to adopt pixel adaptation for object detection and combine it with pseudo-labeling. More recently the Div-Match approach [28] re-elaborated the idea of domain randomization [45]: multiple CycleGAN [53] applications with different constraints produce three extra source variants with which the target can be aligned at different extent through an adversarial multi-domain discriminator. A weak self-training procedure (WST) to reduce false negatives is combined with adversarial background score regularization (BSR) in [27]. Finally, [26] followed the pseudo-labeling strategy including an approach to deal with noisy annotations.

Adaptive Learning on a Budget There is a wide literature on learning from a limited amount of data, both for classification and detection. However, in case of domain shift, learning on a target budget becomes extremely challenging. Indeed, the standard assumption for adaptive learning is that a large amount of unsupervised target samples are available at training time, so that a source model can capture the target domain style from them and adapt to it.

Only few attempts have been done to reduce the target cardinality. In [36] the considered setting is that of *few-shot supervised domain adaptation*: only a few target samples are available but they are fully labeled. In [3,8] the focus is on *one-shot unsupervised style transfer* with a large source dataset and a single unsupervised target image. These works propose time-costly autoencoder-based methods to generate a version of the target image that maintains its content, but visually resembles the source in its global appearance. Thus the goal is image generation with no discriminative purpose. A related setting is that of *online domain adaptation* where unsupervised target samples are initially scarce but accumulate in time [22,48,34]. In this case target samples belong to a continuous data stream with smooth domain changing, so the coherence among subsequent samples can be exploited for adaptation.

Self-Supervised Learning Despite not-being manually annotated, unsupervised data is rich of structural information that can be learned by self-supervision, *i.e.* hiding a subpart of the data information and then trying to recover it. This procedure is generally indicated as *pretext* task and possible examples are image completion [38], colorization [51,29], relative position of patches [12,37], rotation recognition [17] and many more. Self-supervised learning has been extensively used as an initialization step for scarcely annotated supervised learning settings and very recently [1] has shown with a thorough analysis the potential of self-supervised learning from a single image. Recent works also indicated that self-supervision supports adaptation and generalization when combined with supervised learning in a multi-task framework [6,4,50].

Our approach for cross-domain detection relates to the described scenario of learning on a budget and exploits self-supervised learning to perform one-shot unsupervised adaptation. Specifically with OSHOT we show how to recognize objects and their location on a single target image starting from a pre-trained source model, thus without the need of accessing the source data during testing.

3 Method

Problem Setting We introduce the *one-shot unsupervised cross-domain detection scenario* where our goal is to predict on a single image x^t, with t being any target domain not available at training time, starting from N annotated samples of the source domain $S = \{x_i^s, y_i^s\}_{i=1}^N$. Here the structured labels $y^s = (c, b)$ describe class identity c and bounding box location b in each image x^s, and we aim to obtain y^t that precisely detects objects in x^t despite the domain shift.

OSHOT strategy To pursue the described goal, our strategy is to train the parameters of a detection learning model such that it can be ready to get the maximal performance on a single unsupervised sample from a new domain after few gradient update steps on it. Since we have no ground truth on the target sample, we implement this strategy by learning a representation that exploits inherent data information as that captured by a *self-supervised* task, and then finetune it on the target sample (see Figure 2). Thus, we design our OSHOT to include (1) an initial *pretraining* phase where we extend a standard deep detection model adding an image rotation classifier, and (2) a following *adaptation* stage where the network features are updated on the single target sample by further optimization of the rotation objective. Moreover, we exploit *pseudo-labeling* to focus the auxiliary task on the local object context. A clear advantage of this solution is that we decouple source training from target testing, with no need to access the source data while adapting on the target sample.

Preliminaries We leverage on Faster R-CNN [40] as our base detection model. It is a two-stage detector with three main components: an initial block of convolutional layers, a region proposal network (RPN) and a region-of-interest (ROI)

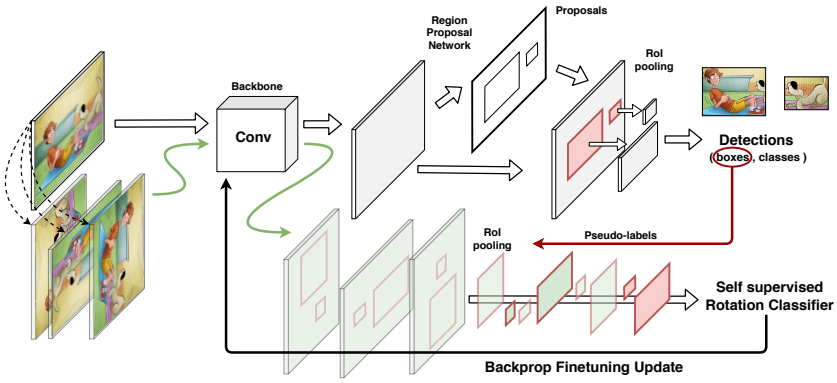


Fig. 2. Visualization of the adaptive phase of OSHOT with cross-task pseudo-labeling. The target image passes through the network and produces detections. While the class information is not used, the identified boxes are exploited to select object regions from the feature maps of the rotated image. The obtained region-specific feature vectors are finally sent to the rotation classifier. A number of subsequent finetuning iterations allows to adapt the convolutional backbone to the domain represented by the test image

based classifier. The bottom layers transform any input image x into its convolutional feature map $G_f(x|\theta_f)$ where θ_f is used to parametrize the feature extraction model. The feature map is then used by RPN to generate candidate object proposals. Finally the ROI-wise classifier predicts the category label from the feature vector obtained using ROI-pooling. The training objective combines the loss of both RPN and ROI, each of them composed by two terms:

$$\begin{aligned}\mathcal{L}_d(G_d(G_f(x|\theta_f)|\theta_d), y) = &\big(\mathcal{L}_{class}(c^*) + \mathcal{L}_{regr}(b)\big)_{RPN} + \\ &\big(\mathcal{L}_{class}(c) + \mathcal{L}_{regr}(b)\big)_{ROI} \ .\end{aligned} \tag{1}$$

Here $\mathcal{L}_{class}$ is a classification loss to evaluate the object recognition accuracy, while $\mathcal{L}_{regr}$ is a regression loss on the box coordinates for better localization. To maintain a simple notation we summarize the role of ROI and RPN with the function $G_d(G_f(x|\theta_f)|\theta_d)$ parametrized by θ_d. Moreover, we use c^* to highlight that RPN deals with a binary classification task to separate foreground and background objects, while ROI deals with the multi-class objective needed to discriminate among c foreground object categories. As mentioned above, ROI and RPN are applied in sequence: they both elaborate on the feature maps produced by the convolutional block, and then influence each other in the final optimization of the multi-task (classification, regression) objective function.

OSHOT pretraining As a first step, we extend Faster R-CNN to include image rotation recognition. Formally, to each source training image x^s we apply four geometric transformations $R(x, \alpha)$ where $\alpha = q \times 90^\circ$ indicates rotations with $q \in \{1, \ldots, 4\}$. In this way we obtain a new set of samples $\{R(x)_j, q_j\}_{j=1}^{M}$

where we dropped the α without loss of generality. We indicate the auxiliary rotation classifier and its parameters respectively with G_r and θ_r and we train our network to optimize the following multi-task objective

$$\underset{\theta_f,\theta_d,\theta_r}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}_d(G_d(G_f(x_i^s|\theta_f)|\theta_d), y_i^s) + \lambda \sum_{j=1}^{M} \mathcal{L}_r(G_r(G_f(R(x^s)_j|\theta_f)|\theta_r), q_j^s) \ , \tag{2}$$

where $\mathcal{L}_r$ is the cross-entropy loss. When solving this problem, we can design G_r in two different ways. Indeed it can either be a Fully Connected layer that naïvely takes as input the feature map produced by the whole (rotated) image $G_r(\cdot|\theta_r) = \mathrm{FC}_{\theta_r}(\cdot)$, or it can exploit the ground truth location of each object with a subselection of the features only from its bounding box in the original map $G_r(\cdot|\theta_r) = \mathrm{FC}_{\theta_r}(\mathit{boxcrop}(\cdot))$. The *boxcrop* operation includes pooling to rescale the feature dimension before entering the final FC layer. In this last case the network is encouraged to focus only on the object orientation without introducing noisy information from the background and provides better results with respect to the whole image option as we will discuss in Section 4.4. In practical terms, both in the case of image and box rotations, we randomly pick one rotation angle per instance, rather than considering all four of them: this avoids any troublesome unbalance between rotated and non-rotated data when solving the multi-task optimization problem.

OSHOT adaptation Given the single target image x^t, we finetune the backbone's parameters θ_f by iteratively solving a self-supervised task on it. This allows to adapt the original feature representation both to the content and to the style of the new sample. Specifically, we start from the rotated versions $R(x^t)$ of the provided sample and optimize the rotation classifier through

$$\underset{\theta_f,\theta_r}{\operatorname{argmin}} \mathcal{L}_r(G_r(G_f(R(x^t)|\theta_f)|\theta_r), q^t) \ . \tag{3}$$

This process involves only G_f and G_r, while the RPN and ROI detection components described by G_d remain unchanged. In the following we use γ to indicate the number of gradient steps (*i.e.* iterations), with $\gamma = 0$ corresponding to the OSHOT pretraining phase. At the end of the finetuning process, the inner feature model is described by θ_f^* and the detection prediction on x^t is obtained by $y^{t*} = G_d(G_f(x^t|\theta_f^*)|\theta_d)$.

Cross-task pseudo-labeling As in the pretraining phase, also in the adaptation stage we have two possible choices to design G_r: either considering the whole feature map $G_r(\cdot|\theta_r) = \mathrm{FC}_{\theta_r}(\cdot)$, or focusing on the object locations $G_r(\cdot|\theta_r) = \mathrm{FC}_{\theta_r}(\mathit{pseudoboxcrop}(\cdot))$. For both variants we include dropout to prevent overfitting on the single target sample. With *pseudoboxcrop* we mean a localized feature extraction operation analogous to that discussed for pretraining, but obtained through a particular form of *cross-task self-training*. Specifically, we follow the self-training strategy used in [27,25] with a cross-task variant: in-

stead of reusing the pseudo-labels produced by the source model on the target to update the detector, we exploit them for the self-supervised rotation classifier. In this way we keep the advantage of the self-training initialization, largely reducing the risks of error propagation due to wrong class pseudo-labels.

More practically, we start from the (θ_f, θ_d) model parameters of the pre-training stage and we get the feature maps from all the rotated versions of the target sample $G_f(\{R(x^t), q\}|\theta_f)$, $q = 1, \ldots, 4$. Only the feature map produced by the original image (*i.e.* $q = 4$) is provided as input to the RPN and ROI network components to get the predicted detection $y^t = (c, b) = G_d(G_f(x^t|\theta_f)|\theta_d)$. This pseudo-label is composed by the class label c and the bounding box location b. We discard the first and consider only the second to localize the region containing an object in all the four feature maps, also recalibrating the position to compensate for the orientation of each map. Once passed through this *pseudoboxcrop* operation, the obtained features are used to finetune the rotation classifier, updating the bottom convolutional network block.

4 Experiments

4.1 Datasets

Real-World (VOC) Pascal-VOC [13] is the standard real-world image dataset for object detection benchmarks. VOC2007 and VOC2012 both contain bounding boxes annotations of 20 common categories. VOC2007 has 5011 images in the train-val split and 4952 images in the test split, while VOC2012 contains 11540 images in the train-val split.

Artistic Media Datasets (AMD) Clipart1k, Comic2k and Watercolor2k [25] are three object detection datasets designed for benchmarking Domain Adaptation methods when the source domain is Pascal-VOC. Clipart1k shares its 20 categories with Pascal-VOC: it has 500 images in the training set and 500 images in the test set. Comic2k and Watercolor2k both have the same 6 classes (a subset of the 20 classes of Pascal-VOC), and 1000-1000 images in the training-test splits each.

Cityscapes [9] is an urban street scene dataset with pixel level annotations of 8 categories. It has 2975 and 500 images respectively in the training and validation splits. We use the instance level pixel annotations to generate bounding boxes of objects, as in [7].

Foggy Cityscapes [43] is obtained by adding different levels of synthetic fog to Cityscapes images. We only consider images with the highest amount of artificial fog, thus training-validation splits have 2975-500 images respectively.

KITTI [16] is a dataset of images depicting several driving urban scenarios. By following [7], we use the full 7481 images for both training (when used as source) and evaluation (when used as target).

Social Bikes is our new concept-dataset containing 30 images of scenes with persons/bicycles collected from Twitter, Instagram and Facebook by searching for *#bike* tags. Square crops of the full dataset are presented in Figure 3: images

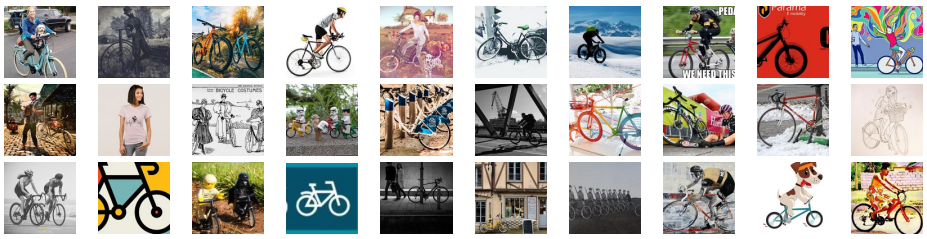


Fig. 3. The Social Bikes concept-dataset. A random data acquisition from multiple users/feeds leads to a target distribution with several, uneven domain shifts

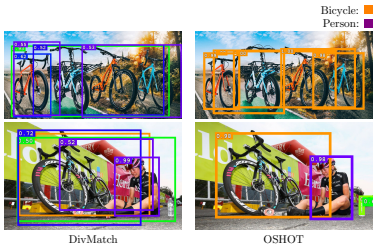
acquired randomly from social feeds show diverse style properties and cannot be grouped under a single shared domain.

4.2 Performance analysis

Experimental Setup We evaluate OSHOT on several testbeds using the described datasets. In the following we will use an arrow $Source \rightarrow Target$ to indicate the experimental setting. Our base detector is Faster-RCNN [35] with a ResNet-50 [21] backbone pre-trained on ImageNet, RPN with 300 top proposals after non-maximum-supression, anchors at three scales (128, 256, 512) and three aspect ratios (1:1, 1:2, 2:1). For all our experiments we set the IoU threshold at 0.5 for the mAP results, and report the average of three independent runs.
OSHOT pretraining. We always resize the image's shorter size to 600 pixels and apply random horizontal flipping. Unless differently specified, we train the base network for 70k iterations using SGD with momentum set at 0.9, the initial learning rate is 0.001 and decays after 50k iterations. We use a batch size of 1, keep batch normalization layers fixed for both pretraining and adaptation phases and freeze the first 2 blocks of ResNet50. The weight of the auxiliary task is set to $\lambda = 0.05$.
OSHOT adaptation. We increase the weight of the auxiliary task to $\lambda = 0.2$ to speed up adaptation and keep all other training hyperparameters fixed. For *each* test instance, we finetune the *initial* model on the auxiliary task for 30 iterations before testing.
Benchmark methods. We compare OSHOT with the following algorithms. *FR-CNN*: baseline Faster-RCNN with ResNet50 backbone, trained on the source domain and deployed on the target without further adaptation. *DivMatch* [28]: cross-domain detection algorithm that, by exploiting target data, creates multiple randomized domains via CycleGAN and aligns their representations using an adversarial loss. *SW* [42]: adaptive detection algorithm that aligns source and target features based on global context similarity. For both DivMatch and SW, we use a ResNet-50 backbone pretrained on ImageNet for fair comparison. Since all cross-domain algorithms need target data in advance and are not designed to work in our one-shot unsupervised setting, we provide them with the advantage of 10 target images accessible during training and randomly selected at each

Table 1. (left) VOC → Social Bikes mAP results; (right) visualization of DivMatch and OSHOT detections. The number associated with each bounding box indicates the model's confidence in localization. Examples show how OSHOT detection is accurate, while most DivMatch boxes are false positives

One-Shot *Target*			
Method	person	bicycle	mAP
FRCNN	67.7	56.6	62.1
OSHOT ($\gamma = 0$)	72.1	52.8	62.4
OSHOT ($\gamma = 30$)	69.4	59.4	**64.4**
Full *Target*			
DivMatch [28]	63.7	51.7	57.7
SW [42]	63.2	44.3	53.7



run. We collect average precision statistics during inference under the favorable assumption that the target domain will not shift after deployment.

Adapting to social feeds When data is collected from multiple sources, the assumption that all target images originate from the same underlying distribution does not hold and standard cross-domain detection methods are penalized regardless of the number of seen target samples. We pretrain the source detector on Pascal VOC, and deploy it on Social Bikes. We consider only the bicycle and person annotations for this target, since all other instances of VOC classes are scarce. We report results in Table 1. OSHOT outperforms all considered competitors, with a mAP score of 64.4. Despite granting them access to the full target, adaptive algorithms incur in negative transfer due to data scarcity and large variety of target styles.

Large distribution shifts Artistic images are difficult benchmarks for cross-domain methods. Unpredictable perturbations in shape and color are challenging to detectors trained only on realistic images. We investigate this setting by training the source detector on Pascal VOC an deploying it on Clipart, Comic and Watercolor datasets. Table 2 summarizes results on the three adaptation splits. We can see how OSHOT with 30 finetuning iterations outperforms all competitors, with mAP gains ranging from 7.5 points on Clipart to 9.2 points on Watercolor. Cross-detection methods perform poorly in this setting, despite using 9 more samples in the adaptation phase compared to OSHOT that only uses the test sample. These results confirm that they are not designed to tackle data scarcity conditions and exhibit negligible improvements compared to the baseline.

Adverse weather Some peculiar environmental conditions, such as fog, may be disregarded in source data acquisition, yet adaptation to these circumstances is crucial for real world applications. We assess the performance of OSHOT on Cityscapes → FoggyCityscapes. We train our base detector on Cityscapes for 30k iterations without stepdown, as in [5]. We select the best performing model on

Table 2. 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*																					
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 [28]	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 [42]	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

(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 [28]	27.1	12.3	26.2	11.5	13.8	34.0	20.8
SW [42]	21.2	14.8	18.7	12.4	14.9	43.9	21.0

(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 [28]	64.6	44.1	44.6	34.1	24.9	60.0	45.4
SW [42]	66.3	41.1	41.1	30.5	20.5	52.3	42.0

Table 3. mAP results for Cityscapes → FoggyCityscapes

Method	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
***One-Shot** Target*									
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 [28]	27.6	38.1	42.9	17.1	27.6	14.3	14.6	32.8	26.9
SW [42]	25.5	30.8	40.4	21.1	26.1	34.5	6.1	13.4	24.7
***Full** Target*									
DivMatch [28]	32.3	43.5	47.6	23.9	38.0	23.1	27.6	37.2	34.2
SW [42]	31.3	32.1	47.4	19.6	28.8	41.0	9.8	20.1	28.8

the Cityscapes validation split and deploy it to FoggyCityscapes. Experimental evaluation in Table 3 shows that OSHOT outperforms all compared approaches. Without finetuning iterations, performance using the auxiliary rotation task increases compared to the baseline. Subsequent finetuning iterations on the target sample improve these results, and 30 iterations yield models able to outperform the second-best method by 5 mAP. Cross-domain algorithms used in this setting struggle to surpass the baseline (DivMatch) or suffer negative transfer (SW).

Cross-camera transfer Dataset bias between training and testing is unavoidable in practical applications, as for urban scene scenarios collected in different cities and with different cameras. We test adaptation between KITTI and Cityscapes in both directions. For cross-domain evaluation we consider only the label car as standard practice. In Table 4, OSHOT improves by 7 mAP points on KITTI → Cityscapes compared to the FRCNN baseline. DivMatch and SW both show a gain in this split, with SW obtaining the highest mAP of 39.2 in the ten-shot setting. We argue that this is not surprising considering that, as shown in the visualization of Table 4, the Cityscapes images share all a uniform

Table 4. mAP of car class in KITTI/Cityscapes detection experiments

One-Shot Target		
Method	KITTI → Cityscapes	Cityscapes → KITTI
FRCNN	26.5	75.1
OSHOT $\gamma = 0$	26.2	**75.4**
OSHOT $\gamma = 10$	33.2	75.3
OSHOT $\gamma = 30$	**33.5**	75.0
Ten-Shot *Target*		
DivMatch [28]	37.9	74.1
SW [42]	39.2	74.6

Table 5. Comparison between baseline, one-shot syle transfer and OSHOT in the one-shot unsupervised cross-domain detection setting

	FRCNN	BiOST [8]	OSHOT ($\gamma = 30$)
mAP on Clipart100	27.9	29.8	**30.7**
mAP on Social Bikes	62.1	51.1	**64.4**
Adaptation time (seconds per sample)	-	$\sim 2.4 * 10^4$	7.8

visual style. As a consequence, 10 target images may be enough for standard cross-domain detection methods. Despite visual style homogeneity, the diversity among car instances in Cityscapes is high enough for learning a good car detection model. This is highlighted by the results in Cityscapes → KITTI task, for which adaptation performance for all methods is similar, and OSHOT with $\gamma = 0$ obtains the highest mAP of 75.4. The FRCNN baseline on KITTI scores a high mAP of 75.1: in this favorable condition detection doesn't benefit from adaptation.

4.3 Comparison with One-Shot Style Transfer

Although not specifically designed for cross-domain detection, in principle it is possible to apply one-shot style transfer methods as an alternative solution for our setting. We use BiOST [8], the current state-of-the-art method for one-shot transfer, to modify the style of the target sample towards that of the source domain before performing inference. Due to the time-heavy requirements to perform BiOST on each test sample[4], we test it on Social Bikes and on a random subset of 100 Clipart images that we name Clipart100. We compare performance and time requirements of OSHOT and BiOST on these two targets. Speed has been computed on an RTX2080Ti with full precision settings.

Table 5 shows summary mAP results using BiOST and OSHOT. On Clipart100, the baseline FRCNN detector obtains 27.9 mAP. We can see how BiOST is effective in the adaptation from one-sample, gaining 1.9 points over the baseline, however it is outperformed by OSHOT, which obtains 30.7 mAP. On Social Bikes, while OSHOT still outperforms the baseline, BiOST incurs in negative transfer, indicating that it was not able to effectively modify the source's style

[4] To get the style update, BiOST trains of a double-variational autoencoder using the entire source besides the single target sample. As advised by the authors through personal communications, we trained the model for 5 epochs.

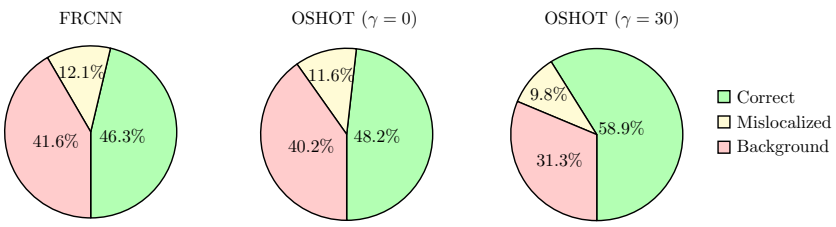


Fig. 4. Detection error analysis on the most confident detections on Clipart

on the images we collected. Furthermore, BiOST is affected by two strong issues: (1) as already mentioned, it has an extremely high time complexity, with more than 6 hours needed to modify the style of a single source instance; (2) it works under the strict assumption of accessing at the same time the entire source training set and the target sample. Due to these weaknesses, and the fact that OSHOT still outperforms BiOST, we argue that existing one-shot translation methods are not suitable for one shot unsupervised cross-domain adaptation.

4.4 Ablation Study

Detection error analysis Following [24], we provide detection error analysis for VOC $\rightarrow$ Clipart setting in Figure 4. We select the 1000 most confident detections, and assign error classes based on IoU with ground truth (IoUgt). Errors are categorized as: correct (IoUgt $\geqslant 0.5$), mislocalized ($0.3 \leqslant$ IoUgt < 0.5) and background (IoUgt < 0.3). Results show that, compared to the baseline FRCNN model, the regularization effect of adding a self-supervised task at training time ($\gamma = 0$) marginally increases the quality of detections. Instead subsequent fine-tuning iterations on the test sample substantially improve the number of correct detections, while also decreasing both false positives and mislocalization errors.

Cross-task pseudo-labeling ablation As explained in Section 3 we have two options in the OSHOT adaptation phase: either considering the whole image, or focusing on pseudo-labeled bounding boxes obtained from the detector after the first OSHOT pretraining stage. For all the experiments presented above we focused on the second case. Indeed by solving the auxiliary task only on objects, we limit the use of background features which may mislead the network towards solutions of the rotation task not based on relevant semantic information (*e.g.*: finding fixed patterns in images, exploiting watermarks). We validate our choice by comparing it against using the rotation task on the entire image in both training and adaptation phases. Table 6 shows results for VOC $\rightarrow$ AMD and Cityscapes $\rightarrow$ Foggy Cityscapes using OSHOT. We observe that the choice of rotated regions is critical for the effectiveness of the algorithm. Solving the rotation task on objects using pseudo-annotations results in mAP improvements that range from 2.9 to 5.9 points, indicating that we learn better features for the main task.

Table 6. Rotating image vs rotating objects via pseudo-labeling on OSHOT

	$G_r(image)$	$G_r(pseudoboxcrop)$
VOC → Clipart	31.0	**33.9**
VOC → Comic	21.0	**26.9**
VOC → Watercolor	48.2	**52.0**
Cityscapes → Foggy Cityscapes	27.7	**31.9**

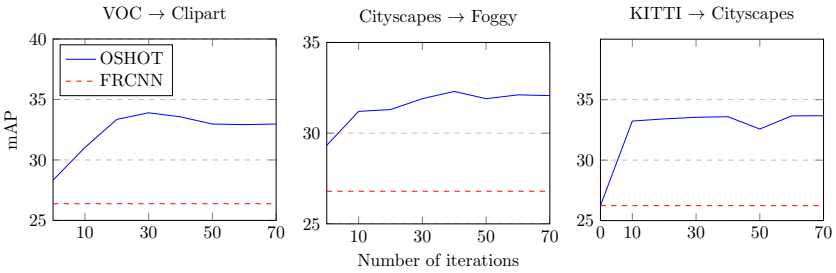


Fig. 5. Performance of OSHOT at different self-supervised iterations

Self-supervised iterations We study the effects of adaptating with up to $\gamma = 70$ iterations on VOC → Clipart, Cityscapes → FoggyCityscapes and KITTI → Cityscapes. Results are shown in Figure 5. We observe a positive correlation between number of finetuning iterations and final mAP of the model in the earliest steps. This correlation is strong for the first 10 iterations and gets to a plateau after about 30 iterations: increasing γ beyond this point doesn't affect the final results.

5 Conclusions

This paper introduced the *one-shot unsupervised cross-domain detection* scenario, which is extremely relevant for monitoring image feeds on social media, where algorithms are called to adapt to a new visual domain from one single image. We showed that existing cross-domain detection methods suffer in this setting, as they are all explicitly designed to adapt from far larger quantities of target data. We presented OSHOT, the first deep architecture able to reduce the domain gap between source and target distribution by leveraging over one single target image. Our approach is based on a multi-task structure that exploits self-supervision and cross-task self-labeling. Extensive quantitative experiments and a qualitative analysis clearly demonstrate its effectiveness.

Acknowledgements This work was partially founded by the ERC grant 637076 RoboExNovo (AD, FCB, SB, BC) and took advantage of the GPU donated by NVIDIA (Academic Hardware Grant, TT). We acknowledge the support provided by Tomer Cohen and Kim Taekyung on their code respectively of BiOST and DivMatch.

References

1. Asano, Y.M., Rupprecht, C., Vedaldi, A.: A critical analysis of self-supervision, or what we can learn from a single image. In: ICLR (2020)
2. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.: A theory of learning from different domains. Machine Learning **79**, 151–175 (2010)
3. Benaim, S., Wolf, L.: One-shot unsupervised cross domain translation. In: NIPS (2018)
4. Bucci, S., D'Innocente, A., Tommasi, T.: Tackling partial domain adaptation with self-supervision. In: ICIAP (2019)
5. Cai, Q., Pan, Y., Ngo, C.W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: CVPR (2019)
6. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: CVPR (2019)
7. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: CVPR (2018)
8. Cohen, T., Wolf, L.: Bidirectional one-shot unsupervised domain mapping. In: ICCV (2019)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
10. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NIPS (2016)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
12. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (2010)
14. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
15. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR **17**(1), 2096–2030 (2016)
16. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
17. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
18. Girshick, R.: Fast r-cnn. In: ICCV (2015)
19. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
22. Hoffman, J., Darrell, T., Saenko, K.: Continuous manifold based adaptation for evolving visual domains. In: CVPR (2014)
23. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: ICML (2018)

24. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV (2012)
25. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018)
26. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: ICCV (2019)
27. 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)
28. 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)
29. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: CVPR (2017)
30. Liu, S., Huang, D., et al.: Receptive field block net for accurate and fast object detection. In: ECCV (2018)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
32. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML (2015)
33. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML (2017)
34. Mancini, M., Karaoguz, H., Ricci, E., Jensfelt, P., Caputo, B.: Kitting in the wild through online domain adaptation. In: IROS (2018)
35. Massa, F., Girshick, R.: maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark` (2018), accessed: 22/08/2019
36. Motiian, S., Jones, Q., Iranmanesh, S., Doretto, G.: Few-shot adversarial domain adaptation. In: NIPS (2017)
37. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
38. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
39. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
40. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS (2015)
41. Russo, P., Carlucci, F.M., Tommasi, T., Caputo, B.: From source to target and back: symmetric bi-directional adaptive gan. In: CVPR (2018)
42. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: CVPR (2019)
43. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. IJCV **126**(9), 973–992 (2018)
44. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV 2016 Workshops (2016)
45. Tobin, J., Fong, R.H., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS (2017)
46. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Adversarial discriminative domain adaptation. In: CVPR (2017)

47. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
48. Wulfmeier, M., Bewley, A., Posner, I.: Incremental adversarial domain adaptation for continually changing environments. In: ICRA (2018)
49. Xie, R., Yu, F., Wang, J., Wang, Y., Zhang, L.: Multi-level domain adaptive learning for cross-domain detection. In: ICCV Workshops (2019)
50. Xu, J., Xiao, L., López, A.M.: Self-supervised domain adaptation for computer vision tasks. ArXiv **abs/1907.10915** (2019)
51. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
52. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: CVPR (2018)
53. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)