

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/340963079>

Unsupervised Domain Adaptation with Multiple Domain Discriminators and Adaptive Self-Training

Preprint · April 2020

CITATIONS

0

READS

32

4 authors, including:



[Marco Toldo](#)

University of Padova

5 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)



[Umberto Michieli](#)

University of Padova

16 PUBLICATIONS 47 CITATIONS

[SEE PROFILE](#)



[Pietro Zanuttigh](#)

University of Padova

110 PUBLICATIONS 1,288 CITATIONS

[SEE PROFILE](#)

Unsupervised Domain Adaptation with Multiple Domain Discriminators and Adaptive Self-Training

Teo Spadotto, Marco Toldo, Umberto Michieli and Pietro Zanuttigh
 Department of Information Engineering, University of Padova, 35131, Padova, Italy

Abstract—Unsupervised Domain Adaptation (UDA) aims at improving the generalization capability of a model trained on a source domain to perform well on a target domain for which no labeled data is available. In this paper, we consider the semantic segmentation of urban scenes and we propose an approach to adapt a deep neural network trained on synthetic data to real scenes addressing the domain shift between the two different data distributions. We introduce a novel UDA framework where a standard supervised loss on labeled synthetic data is supported by an adversarial module and a self-training strategy aiming at aligning the two domain distributions. The adversarial module is driven by a couple of fully convolutional discriminators dealing with different domains: the first discriminates between ground truth and generated maps, while the second between segmentation maps coming from synthetic or real world data. The self-training module exploits the confidence estimated by the discriminators on unlabeled data to select the regions used to reinforce the learning process. Furthermore, the confidence is thresholded with an adaptive mechanism based on the per-class overall confidence. Experimental results prove the effectiveness of the proposed strategy in adapting a segmentation network trained on synthetic datasets like GTA5 and SYNTHIA, to real world datasets like Cityscapes and Mapillary.

I. INTRODUCTION

Deep neural networks are nowadays the most common solution to learn complex tasks from large amounts of training data. Unfortunately, they can be extremely weak in generalizing the learned knowledge to new domains, as even a small domain shift between train and test data distributions could cause serious harm to the accuracy of the network. This is particularly relevant if we aim at transferring knowledge acquired on synthetic data to the real world. Indeed, a large number of applications would benefit from training deep networks on large amounts of synthetic data. An example are autonomous driving systems, that should be able to understand the surrounding environment from visual data. In case of road scenes, the pixel-level annotation must be manually provided for a huge amount of frames acquired by cameras mounted on moving cars, which is highly expensive, time-consuming and error-prone. For this reason, synthetic imagery collected from graphics game engines could be extremely useful for this process [1], [2]. However, despite the great realism of such scenes, a domain shift exists between computer generated data and real world images. Without additional provisions, a deep network trained on synthetic images would provide low accuracy on real data.

Our work was in part supported by the Italian Minister for Education (MIUR) under the “Departments of Excellence” initiative (Law 232/2016).

To address this issue, we developed an UDA strategy for road driving scenes to adapt an initial learning performed on synthetic data to the real world case. The complete framework presented in this work is made of four components. First, a standard cross-entropy loss is employed to perform a supervised training on synthetic data with ground truth annotations. Then, two adversarial learning schemes with a couple of fully convolutional discriminators are jointly employed. We start from the structure used in [3], [4] for semi-supervised semantic segmentation and adapted in [5], [6] to Unsupervised Domain Adaptation (UDA). However, in this work we use two discriminative networks, one to differentiate between ground truth and predicted segmentations and the other to discriminate between segmentation maps coming from synthetic and real world data. Finally, a self-training component is introduced based on the idea that the output of the discriminator provides a measure of the reliability of the network estimations to be exploited in a self-training framework [3], [5]. In previous works, this module lacked two fundamental aspects: it was not class-wise adaptive and was not mutable during training. In other words, different classes shared the same confidence threshold to select regions for self-training, and its value was kept fixed during all the learning process. In this work, instead, we introduce an adaptive thresholding scheme for the selection of the regions used for self-training that enforces a balanced selection for all classes and allows for variability over different training steps. Hence, the framework can accomplish both inter-class confidence flexibility and time adaptability throughout the optimization phase. We test our model on the task of domain adaptation for semantic segmentation of urban scenes from synthetic to real world domains. In particular, we employ the SYNTHIA and GTA5 synthetic datasets to train the supervised component, whereas we resort to real data from Cityscapes and Mapillary datasets for the unsupervised adaptation modules.

In summary, the main contributions of this work are: (1) we introduce a novel adversarial scheme exploiting multiple domain discriminators to align the source and target domains, (2) we design a self-training module with adaptive confidence both over different classes and over different training steps, (3) we prove the effectiveness of our framework on 4 different experimental scenarios outperforming competing approaches.

II. RELATED WORK

Semantic Segmentation of images has recently witnessed substantial improvements thanks to deep learning architectures.

State-of-the-art approaches for this task typically rely on an encoder-decoder structure, allowing the extraction of global semantic clues while retaining input spatial dimensionality. Starting from the well-known FCN architecture [7], many models have been proposed, such as the PSPNet [8], DRN [9] and DeepLab in its multiple versions [10]–[12] (in our work we resort to the DeepLab-v2 model [10] as the base segmentation network). The main issue affecting these highly-performing models is the high complexity, making their success strictly related to the availability of massive amounts of labelled data. For this reason, many datasets have been created (e.g., Cityscapes [13] or Mapillary [14] for urban scenes), even though the pixel-wise annotation procedure is highly expensive and time consuming. In light of this, recent research has studied ways to exploit knowledge extracted from other sources where labels are easily accessible, and data shares as much as possible the same distribution of the target domain.

Domain Adaptation, especially in its unsupervised form, has drawn wide attention in the research community, as it represents a possible solution to the data shortage problem affecting several visual recognition tasks. Early works on domain adaptation for deep networks mainly focus on matching the statistical distribution of network embeddings, promoting domain invariant latent representations, while a task-specific loss preserves feature discriminativeness. The distribution alignment is achieved by estimating and minimizing a measure of domain discrepancy, as the Maximum Mean Discrepancy (MMD) [15]–[17] or a correlation distance [18], [19].

The introduction of adversarial learning [20] has given rise to new adaptation techniques, which make use of an additional domain classifier to bridge the domain gap between source and target data distributions. The adversarial approach has been widely investigated in classification tasks, being applied both on the latent space, to align source and target representations [21]–[23], and within the input space, to produce a form or target supervision provided by labeled target-like source samples [24]–[26]. Recently, a multitude of diverse adaptation strategies have been proposed to bridge the source-target domain discrepancy. Some examples include feature clustering for class-conditional alignment [27], [28], minimization of a sliced Wasserstein distance unaffected by class imbalance [29], feature norm matching [30], regularization provided by a joint source-target predictor [31], entropy minimization-maximization for cross-domain feature alignment [32].

Unlike image classification tasks, semantic segmentation involves dense highly-structured predictions and pixel-level precision in the output space is required. Hence, the inherent higher complexity of network representations to capture both global and local semantics demands more sophisticated techniques to address the domain adaptation task. Multiple methods based on adversarial learning have been proposed, focusing on distribution alignment in the input space [33], feature space [34], output space [35], [36] or more effectively on a combination of those [37]–[42]. In alternative to adversarial learning, several approaches have been experimented, such as adversarial dropout [43]–[45], entropy minimization [46],

[47], style transfer for generation of artificial target supervision [48]–[50] and curriculum-style learning [51], [52].

Self-Training is another tool that can be exploited: highly-confident predictions on unlabeled data are used to generate pseudo-labels, which are then used to train the classifier with a self-generated supervision. While commonly employed in semi-supervised learning [3], [53], self-training techniques have been recently adopted to address the domain adaptation task, as concurrently learning from source annotations and target pseudo-labels intrinsically promotes domain alignment over network embeddings [5], [6], [52], [54], [55]. An issue when applying self-teaching to semantic segmentation is that different classes show very different distributions and frequencies, as well as variable transferability across domains. Hence, a simple class-unaware confidence-based pseudo-label filtering leads towards the optimization of the easy classes, whereas the harder ones are neglected, thus hindering the process of knowledge transfer to the unlabeled domain. For this reason, effective self-training approaches rely on class balancing techniques, based on, for example, class weights computed *a priori* over source annotations [5], [6] or extrapolated from the confidence statistic of all the pixels in the target set [54], [55]. In our approach we opt for a more effective solution, as we perform class-wise confidence threshold selection at each training step by looking at network predictions on target data in the current batch, thus avoiding assumptions on shared source-target properties, or slow computations involving the analysis of the whole employed dataset.

III. PROPOSED DOMAIN ADAPTATION STRATEGY

In this section the complete architecture of the proposed approach is discussed. An overview of the method is given in Figure 1. We denote with G the semantic segmentation network that we want to adapt from supervised synthetic data to unlabeled real data. G takes the role of the generator in our adversarial setup. The optimization of the segmentation network is driven by the minimization of a multi-target objective involving four loss functions. In particular, to guide the adaptation of G to unlabeled real data we employ two discriminative networks and a self-training module.

Let us denote with \mathbf{X}_n^s the generic n -th image in the source (synthetic) domain and with \mathbf{Y}_n^s the corresponding ground truth segmentation, while \mathbf{X}_n^t is the n -th real world sample (for which ground truth is not available during training). The first component of our approach, $\mathcal{L}_{G,0}$, is a standard cross-entropy loss working on labeled synthetic data.

The second and the third loss functions, minimized during the training of G , are relative to the adversarial training with the couple of discriminator networks. The first discriminator module D_1 is trained to distinguish between ground truth and generated maps (the latter either coming from synthetic or real images). The peculiarity of this network is that it is a fully convolutional model and it produces a per-pixel confidence estimation, differently from traditional adversarial frameworks where the discriminator outputs a single binary value for the whole input image. The key idea is that the discriminative

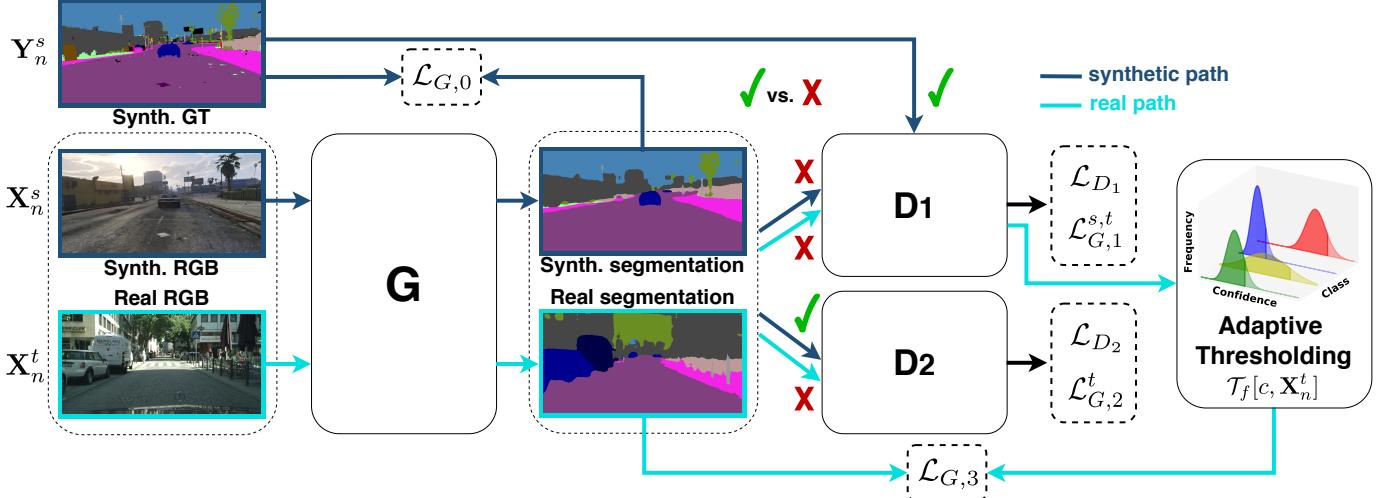


Fig. 1: Architecture of the proposed approach (*best viewed in colors*). The semantic segmentation network G is trained with the combination of 4 losses: a supervised cross entropy on source data $\mathcal{L}_{G,0}$, a double adversarial framework $\mathcal{L}_{G,1}^{s,t}$ and $\mathcal{L}_{G,2}^t$, and a self-training module $\mathcal{L}_{G,3}$ with class-wise and time-varying adaptive thresholding mask \mathcal{T}_f .

action provides a measure of discrepancy between the statistic of source annotations and both source and target prediction maps, and its optimization leads to an indirect yet effective cross-domain distribution alignment [3], [5], [6].

The discriminator network D_1 is made of a stack of 5 convolutional layers each using 4×4 kernels with a stride of 2 and Leaky ReLU activation function. The number of filters (from first to last layer) is 64, 64, 128, 128, 1 and the cascade is followed by a bilinear upsampling to match the original input image resolution. The network D_1 is trained to minimize the loss \mathcal{L}_{D_1} that is a standard cross-entropy loss between D_1 's output and the one-hot encoding indicating whether the input segmentation map is the ground truth (class 1) or it is produced by G (class 0), i.e.:

$$\mathcal{L}_{D_1} = - \sum_{p \in \mathbf{X}_n^{s,t}} \log(1 - D_1(G(\mathbf{X}_n^{s,t}))^{(p)}) + \log(D_1(\mathbf{Y}_n^s)^{(p)}) \quad (1)$$

where p is a generic pixel in the image. Meanwhile, G is forced to produce segmentation maps that resemble ground-truth annotations when inspected by D_1 :

$$\mathcal{L}_{G,1}^{s,t} = - \sum_{p \in \mathbf{X}_n^{s,t}} \log(D_1(G(\mathbf{X}_n^{s,t}))^{(p)}) \quad (2)$$

The second discriminator, D_2 , is trained to differentiate between segmentation maps coming from synthetic or real-world data. Differently from D_1 , D_2 is always fed with the generator output, i.e. with source or target images segmented by G , whereas no ground-truth information is employed. D_2 has the same structure as D_1 except that its number of channels has been reduced to 48, 48, 96, 96, 1, as its adaptation objective is complementary to the one of D_1 and requires less computational complexity to be accomplished. The adversarial loss for D_2 can be expressed as:

$$\mathcal{L}_{D_2} = - \sum_{p \in \mathbf{X}_n^{s,t}} \log(1 - D_2(G(\mathbf{X}_n^t))^{(p)}) + \log(D_2(G(\mathbf{X}_n^s))^{(p)}) \quad (3)$$

which is a standard cross-entropy loss similar to \mathcal{L}_{D_1} . Here the objective is formulated between D_2 's output and the one-hot encoding marking with 0 and 1 the segmentation maps produced by G from respectively target and source images. The role of G in the second adversarial competition is to fool D_2 , by computing sufficiently realistic target prediction maps resembling the source domain ones. Hence, similarly to Eq. 2, the adversarial loss for G related to D_2 is of the form:

$$\mathcal{L}_{G,2}^t = - \sum_{p \in \mathbf{X}_n^t} \log(D_2(G(\mathbf{X}_n^t))^{(p)}) \quad (4)$$

The second adversarial framework is specifically focused on the adaptation of target network representations, in that G is trained to produce target segmentation maps which are close to source ones from a statistical point of view. Thus it directly tackles the domain gap causing the performance drop on the target set. Source-target alignment also happens within the first adversarial scenario as a side effect. Indeed, G 's output is forced to be distributed as ground-truth labels for both source and target inputs, but no actual cross-domain adaptation of network embeddings is directly performed.

The last module in our adaptation framework implements a self-training strategy. As shown by recent works [3], [5], [6], the output of the discriminator D_1 can be interpreted as a measure of confidence for predicted target pixels, since D_1 should identify target predictions that deviate from source ground-truth statistic. Hence, the output of D_1 can be exploited to select the reliable predictions in the target segmentation maps. The selected output samples are then converted into one-hot encoded data, which can be used as a self-taught ground-truth to train G directly on unlabeled target samples. The self-training objective is expressed by the following loss:

$$\mathcal{L}_{G,3} = - \sum_{p \in \mathbf{X}_n^t} \sum_{c \in \mathcal{C}} \mathcal{M}_f^{(p)} \cdot W_c^s \cdot \hat{\mathbf{Y}}_n^{(p)}[c] \cdot \log(G(\mathbf{X}_n^t)^{(p)}[c]) \quad (5)$$

where $\hat{\mathbf{Y}}_n$ is the one-hot encoded ground truth derived from the per-class argmax of the generated probability map $G(\mathbf{X}_n^t)$, c is a specific class belonging to the set of possible classes \mathcal{C} and W_c^s is a weighting function proportional to the class frequency on the source domain as introduced in [5], [6]. $\mathcal{M}_f^{(p)}[c, \mathbf{X}_n^t]$ (for ease of notation we drop its dependencies on class and training sample in all the equations) is the adaptive thresholding mask for the selection of the regions used for self-training, defined as:

$$\mathcal{M}_f^{(p)} = \begin{cases} 1 & \text{if } (D_1(G(\mathbf{X}_n^t))^{(p)} > \mathcal{T}_f[c, \mathbf{X}_n^t]) \wedge (\hat{\mathbf{Y}}_n^{(p)}[c] = 1) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Differently from previous works [5], [6], [52], [55], in which the confidence threshold was computed before-hand and kept fixed throughout the training, here we propose to have both a class-level and a training-stage adaptation of the threshold. Indeed, different classes typically have different confidence values which may also vary during training. The class-wise confidence threshold selection for a generic training step and a generic class c is formulated as:

$$\mathcal{T}_f[c, \mathbf{X}_n^t] = \mathcal{Q}_f(D_1(G(\mathbf{X}_n^t)[c])) \quad (7)$$

where \mathcal{Q}_f represents the f -th percentile. The best results are obtained with f in the range of 75–80% to enable self-training only in regions with high confidence on target data. We compute \mathcal{T}_f for every class at each training step by looking at network predictions on target data in the current batch. This makes the model adaptive both to the statistic of the various classes and to the different training phases. Finally, we can compute the overall loss function for G with a weighted average of the four individual losses, i.e.:

$$\mathcal{L}_{full} = \mathcal{L}_{G,0} + w_1^{s,t} \mathcal{L}_{G,1}^{s,t} + w_2^t \mathcal{L}_{G,2}^t + w_3 \mathcal{L}_{G,3} \quad (8)$$

with weighting parameters $w_1^{s,t}$, w_2^t and w_3 empirically set.

IV. EXPERIMENTAL RESULTS

A. Datasets

We evaluate our framework on some publicly available and widely used datasets. The supervised synthetic training is performed on the GTA5 [1] and SYNTHIA [2] datasets. The real world datasets for unsupervised adaptation and for results evaluation are Cityscapes [13] and Mapillary [14]. The evaluation scenario is the same of competing approaches as [5], [6], [34], [39], [51] to allow for a fair comparison. We provide the segmentation network with images of 750×375 px during training for memory constraints, while testing is done at the original resolution.

GTA5 [1] contains 24966 synthetic images acquired through the video game *Grand Theft Auto 5* and labeled into 19 classes compatible with real datasets. The images are taken from the car perspective in American-style virtual cities and are characterized by a high level of quality and realism. We used 23966 images for supervised training while 1000 have been taken out for validation purposes.

SYNTHIA [2] contains 9400 synthetic images rendered with an ad-hoc engine (we used the *SYNTHIA-RAND-CITYSCAPES* subset). Frames come from virtual European-style towns with a large variability of scenes under various light and weather conditions. On the other hand, the visual quality is lower than that of GTA5. For the evaluation on real datasets, only 16 overlapping classes are taken into consideration. We used 9300 images for supervised training and 100 for validation purposes.

Cityscapes [13] contains 2975 images acquired in 50 European cities. The original training set (without labels) has been used for unsupervised adaptation, while the 500 images in the original validation set have been used as a test set (as done by competing approaches since the test set labels are not publicly available).

Mapillary [14] contains 20000 high resolution images, whose diversity in terms of semantic categories and acquisition settings ensure improved data expressiveness, essential to address the semantic segmentation task. We employed all 18000 images from the training set (without labels) when performing unsupervised adaptation in the training stage, while we resorted to the validation set of 2000 images for testing, as done by competing approaches. Following [56], we perform class matching to the Cityscapes dataset.

B. Training Details

The approach is agnostic to the deep learning architecture used for G : in principle any semantic segmentation network can be used, in our case we employ the well known Deeplab v2 model [10] with the ResNet-101 backbone. The weights are pre-trained [57] on the MSCOCO dataset [58].

The proposed deep learning scheme is implemented in TensorFlow and the code will be available soon. The generator network G is trained as proposed in [10] using the Stochastic Gradient Descent (SGD) optimizer with momentum set to 0.9 and weight decay to 10^{-4} . The discriminators D_1 and D_2 are trained using the Adam optimizer. The learning rate employed for both G and D_1 starts from 10^{-4} and is decreased up to 10^{-6} by means of a polynomial decay with power 0.9. As for D_2 , the base learning rate is set to 10^{-4} and 5×10^{-4} for respectively the SYNTHIA and GTA5 adaptation scenarios. The weighting parameters are empirically set as $w_1^s = 10^{-2}$, $w_1^t = 10^{-3}$ and $w_2^t = 10^{-2}$ independently of the source and target datasets. The self-training parameter w_3 is fixed to 5×10^{-2} and 10^{-1} for respectively SYNTHIA and GTA5 cases, as the inferior realism of the SYNTHIA dataset suggests a more cautious usage of the self-training module. We train the model for 20K iterations on a NVIDIA RTX 2080 Ti GPU. The longest training inside this work, i.e., the one with all the loss components enabled, takes about 6 hours to complete.

C. Evaluation on the Cityscapes Dataset

The first scenario we consider for evaluation purposes comprises the adaptation to the Cityscapes dataset from both SYNTHIA and GTA5. As done by competing approaches [34], [35], [59], the numerical performance is expressed in terms of mean Intersection over Union (mIoU) between predicted

		Method	road	sidewalk	building	wall	fence	pole	t light	t sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mean
a)	To Cityscapes	Supervised ($\mathcal{L}_{G,0}$ only)	49.3	24.4	56.4	6.5	19.6	25.6	23.6	10.1	82.7	28.5	69.9	55.5	4.9	80.9	18.0	33.0	1.2	15.1	0.1	31.9
		Hoffman et al. [34]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
		Hung et al. [3]	81.7	0.3	68.4	4.5	2.7	8.5	0.6	0.0	82.7	21.5	67.9	40.0	3.3	80.7	34.2	45.9	0.2	8.7	0.0	29.0
		Zhang et al. [51]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	14.6	28.9
		Biasetton et al. [5]	54.9	23.8	50.9	16.2	11.2	20.0	3.2	0.0	79.7	31.6	64.9	52.5	7.9	79.5	27.2	41.8	0.5	10.7	1.3	30.4
		Michieli et al. [6]	81.0	19.6	65.8	20.7	2.9	20.9	6.6	0.2	82.4	33.0	68.2	54.9	6.2	80.3	28.1	41.6	2.4	8.5	0.0	33.3
		Ours	77.7	35.9	67.2	18.9	12.1	26.2	15.9	5.9	83.7	33.3	72.7	53.9	4.2	82.6	21.5	41.1	0.1	13.9	0.0	35.1
b)	To SYNTHIA	Supervised ($\mathcal{L}_{G,0}$ only)	17.9	24.2	38.6	5.0	0.0	28.7	0.0	4.5	79.3	-	80.8	54.0	8.9	75.7	-	35.4	-	4.2	3.9	28.8
		Hoffman et al. [34]	11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	-	68.7	51.2	3.8	54.0	-	3.2	-	0.2	0.6	20.1
		Hung et al. [3]	72.5	0.0	63.8	0.0	0.0	16.3	0.0	0.5	84.7	-	76.9	45.3	1.5	77.6	-	31.3	-	0.0	0.1	29.4
		Zhang et al. [51]	65.2	26.1	74.9	0.1	0.5	10.7	3.7	3.0	76.1	-	70.6	47.1	8.2	43.2	-	20.7	-	0.7	13.1	29.0
		Biasetton et al. [5]	78.4	0.1	73.2	0.0	0.0	16.9	0.0	0.2	84.3	-	78.8	46.0	0.3	74.9	-	30.8	-	0.0	0.1	30.2
		Michieli et al. [6]	80.7	0.3	75.0	0.0	0.0	19.5	0.0	0.4	84.0	-	79.4	46.6	0.8	80.8	-	32.8	-	0.5	0.5	31.3
		Ours	72.0	26.6	66.1	1.8	0.0	30.2	0.0	4.3	81.4	-	82.2	51.7	4.0	82.2	-	37.9	-	7.7	5.9	34.6
c)	To Mapillary	Supervised ($\mathcal{L}_{G,0}$ only)	69.8	31.8	58.8	14.6	22.3	28.3	31.8	28.8	70.0	24.4	72.4	60.4	16.8	80.6	36.6	34.3	10.2	26.2	0.2	37.8
		Hung et al. [3]	78.2	29.7	68.7	10.0	6.7	17.5	0.0	0.0	76.4	35.2	95.6	53.8	13.8	77.5	34.3	30.2	5.0	21.8	0.0	34.4
		Biasetton et al. [5]	71.4	25.0	62.0	20.4	17.6	26.8	5.9	0.8	64.6	24.6	86.5	58.3	14.7	80.0	39.3	42.2	5.5	22.3	0.1	35.2
		Michieli et al. [6]	79.9	28.0	73.4	23.0	29.5	20.9	1.1	0.0	79.5	39.6	95.0	57.6	9.0	80.6	41.5	40.1	7.4	24.8	0.1	38.5
		Ours	80.0	43.3	75.4	19.4	29.7	29.6	23.3	16.2	78.5	33.5	93.7	59.0	20.3	82.2	44.5	43.4	2.5	22.1	0.0	41.9
		Supervised ($\mathcal{L}_{G,0}$ only)	25.4	22.0	56.4	6.9	0.1	29.4	0.0	2.8	72.8	-	92.1	53.7	16.1	75.1	-	30.8	-	8.6	5.8	31.1
d)	To SYNTHIA	Hung et al. [3]	36.8	20.1	53.9	0.0	0.0	23.7	0.0	0.0	73.9	-	95.6	43.4	0.1	64.6	-	19.0	-	0.4	0.5	27.0
		Biasetton et al. [5]	16.4	19.1	42.2	2.7	0.0	33.1	0.0	1.3	76.5	-	88.0	50.4	10.9	69.9	-	25.5	-	6.1	9.2	28.2
		Michieli et al. [6]	57.6	18.3	62.1	0.4	0.0	23.7	0.0	0.0	79.4	-	94.8	52.4	9.2	74.2	-	28.3	-	4.0	6.9	32.0
		Ours	59.0	28.4	68.6	0.7	0.0	29.8	0.0	1.8	77.5	-	94.9	54.6	12.6	76.8	-	28.0	-	11.2	14.7	34.9

TABLE I: Per-class and mean IoU on the four considered UDA scenarios. The approaches have been trained in a supervised way on the synthetic dataset and the unsupervised domain adaptation has been performed using the respective real world training set. The results are reported on the real world validation sets.

maps and ground-truth labels over the Cityscapes validation set. The per-class mIoU results of the evaluations are displayed in Table Ia) and Ib). We denote as *supervised* the naïve approach relying only on source supervision and no form of adaptation, while the numerical performance of our method as a whole is reported in the last row of each section.

Using the GTA5 as source domain, the simple supervised approach achieves a final mIoU of 31.8% on the target dataset. The introduction of the multi-domain adversarial learning scheme and adaptive self-training module leads to a performance increment of 3.3%, boosting the mIoU up to 35.1%. Moreover, the improvement is shared by the majority of the semantic classes. Some categories characterized by semantic similarity between them (such as *road*, *sidewalk* and *terrain*) or with appearance discrepancy across source and target domains (such as *car*, *truck* and *bus*) seem to highly benefit from the adaptation, proving that our method succeeded in bridging the domain gap. In Table I we also report results achieved by some competing approaches. It can be noticed that our strategy outperforms all of them, including those relying on simpler self-training and adversarial learning schemes (i.e., [3], [5], [6]), demonstrating the efficacy of the multi-domain discrimination and of the adaptive thresholding techniques we introduced. In Figure 2a) we show some sample segmented images from the Cityscapes validation set. We can appreciate that our approach leads to a more precise detection of several semantic entities found in the input image. For example, the *road* and *sidewalk* classes are subject to a more accurate

recognition, which supports the numerical results.

Section (b) of Table I reports the results of the adaptation from the SYNTHIA dataset. As for the GTA5 case, our adaptation strategy represents a considerable improvement over the supervised training on the source dataset. The mIoU achieved without adaptation (28.8%) is pushed up to 34.6% by our framework. The increment of almost 6% is even higher than the one achieved with the more realistic GTA5 dataset as source domain, and this proves the effectiveness of the adaptation modules we developed also in a challenging environment with a larger statistical gap across domains. For example, while texture discrepancy between Cityscapes and SYNTHIA causes the road on real-world scenes to be hardly recognized by the segmentation network in lack of adaptation, our strategy successfully provides the predictor with road detection capabilities. Figure 2a) includes some qualitative results. Again, we can observe the improved semantic understanding and detection accuracy on classes such as *road* and *sidewalk* with respect to the baseline, as well as with respect to a competing approach based on adversarial and self-training techniques [5].

D. Evaluation on the Mapillary dataset

We evaluate our approach also on the Mapillary dataset. We start by using the GTA5 dataset for the supervised training as before: the results are shown in Table Ic). With no adaptation to the Mapillary dataset the network achieves a mIoU of 37.8%. Thanks to the multiple domain discriminators and to the adaptive self-training techniques our framework is able to

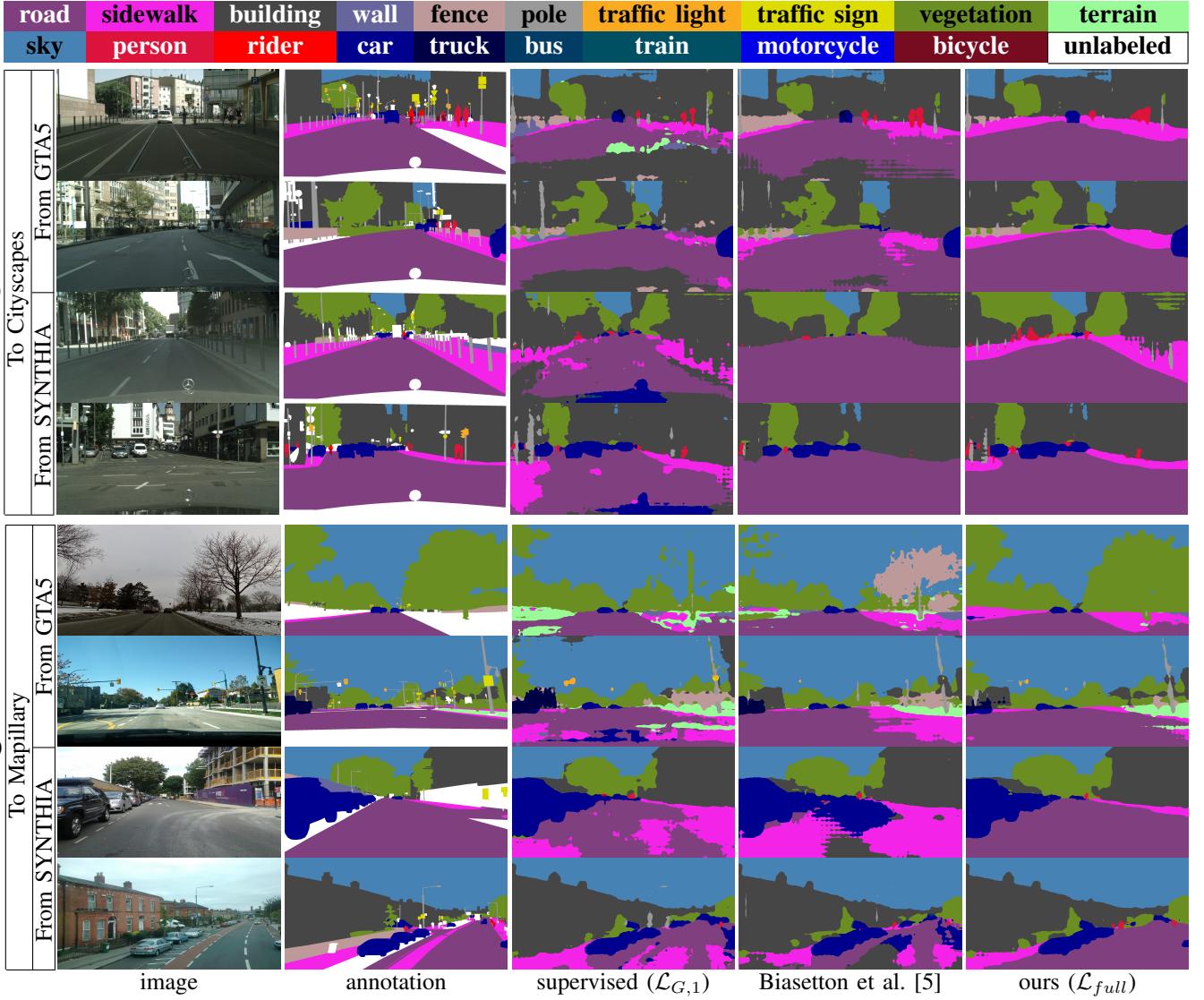


Fig. 2: Semantic segmentation of some sample scenes extracted from the Cityscapes (a) and Mapillary (b) validation datasets. The first group of four rows is related to the Cityscapes dataset, the last four to the Mapillary dataset. For each group, the first two rows are related to the experiments in which the GTA5 dataset is used as source. The last two rows are related to the case in which the SYNTHIA dataset is used as source (*best viewed in colors*).

reach a mIoU of 41.9%, significantly outperforming all the compared methodologies. We can notice that the improvement with respect to the baseline approach is consistently distributed among the semantic classes and it is particularly evident on the *road* or *building* ones. Qualitative results are shown in the first two rows of Fig. 2b), where we can verify that most of the noise present in the supervised training and in [5] is filtered out by the proposed framework. In particular, the *vegetation* and *sidewalk* categories highly benefit from the domain adaptation with class-wise and time-variable confidence threshold selection.

Furthermore, we can appreciate that, also on the Mapillary dataset, the accuracy is lower when the SYNTHIA dataset is used for supervised training, leading to a mIoU of 31.3% only. As already noticed from the evaluations on Cityscapes, some

classes (i.e., *road*, *sidewalk* and *building*) have a low accuracy due to the poor texture representation and vastly profit by the adaptation to the target domain. The complete framework increases the final mIoU to 34.9% with an improvement of 3.8%, consistent with the previous experiments made across different datasets. Remarkable are the percentage gains in the aforementioned classes: for example, the *road* class more than doubles its accuracy and *sidewalk*'s IoU grows by 12.2%. The visual results are reported in the last two rows of Fig. 2b). Here, for example, we can appreciate that the proposed approach is the only one to achieve an accurate and reliable recognition of *road* and *cars* classes on the shown images, which confirms our previous analysis.

$\mathcal{L}_{G,0}$	$\mathcal{L}_{G,1}^s$	$\mathcal{L}_{G,1}^t$	$\mathcal{L}_{G,2}^t$	$\mathcal{L}_{G,3}$	\mathcal{T}_f	mIoU
✓						37.8
✓		✓	✓	✓	✓	39.9
✓	✓		✓	✓	✓	40.3
✓	✓	✓		✓	✓	40.7
✓	✓	✓	✓			41.1
✓	✓	✓	✓	✓		40.6
✓	✓	✓	✓	✓	fix 0.2	40.9
✓	✓	✓	✓	✓	✓	41.9

TABLE II: Ablation results on the Mapillary validation set adapting from GTA5.

E. Ablation Study

In this section we present an accurate investigation of the effectiveness of the various modules of the proposed framework. For this study we consider the performances on the Mapillary dataset when adapting from GTA5. We start by evaluating the individual impact of each module: the performance analysis is shown in Table II. Let us recall that the baseline architecture, i.e., the Deeplab-v2 network trained on synthetic data only, achieves a mIoU of 37.8% on real data. Then, we analyze the remaining modules by removing one component at a time (row 2 to 6). We can appreciate that all the components bring a significant contribution to the final mIoU, which in the full version of the approach where all of them are enabled is 41.9%. The impact of $\mathcal{L}_{G,1}^s$, $\mathcal{L}_{G,1}^t$ and $\mathcal{L}_{G,2}$ is clear by looking at Table II: without each of them the accuracy decreases with respect to the complete framework but remains higher than the source supervised case. In particular, we performed a more detailed analysis of the self-training module. Having no self-training leads to 41.1% of mIoU, while having self-training done on all pixels (without thresholding with \mathcal{T}_f) or using a fixed confidence threshold (e.g., setting a constant value of $\mathcal{T}_f = 0.2$ as in [5]) leads to 40.6% and 40.9%, respectively. These results show that self-training is not effective if the reliable pixels are not accurately selected, e.g., if performed immutable over the classes and the training steps. On the other side we found that self-training is effective with confidence thresholds variable over classes and over training time.

We also analyzed the behavior of the per-class time-varying confidence values: they are shown in Fig. 3 for the GTA5 to Mapillary scenario. Here, we can appreciate how the confidence thresholds vary over the training time and typically converge to a value which may be significantly different among the various classes. While previous works [5], [6] fixed the threshold to 0.2, here we can see how the desired value is variable and ranges between 0.04 and 0.4. Highly confident classes typically have high confidence threshold, in order to propagate back very reliable predictions through self-training as for instance *road*, *sky* and *building*. However, notice that *train* and *bike* have high confidence threshold values only because they are predicted too rarely to be useful for confidence assessment (i.e., the thresholds are piece-wise constant functions), hence represent failure modes of the estimate. More challenging classes such as *wall* and *fence*, instead, are characterized by a much lower confidence value. The most important aspect, however, is to

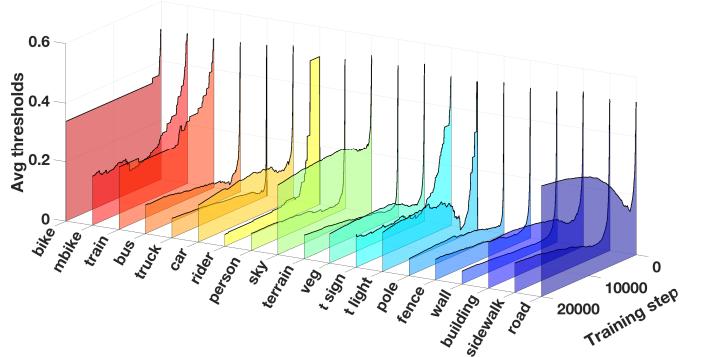


Fig. 3: Time average over the initial to current step interval of per-class confidence thresholds for different classes and at different training steps on the Mapillary dataset when adapting from GTA5 (*best viewed in colors*).

verify that our framework can adapt both to the properties of different classes and to the changes in the network behavior during the training procedure. Additionally, we can compute the mean threshold value averaged over all the classes at the end of the training phase and compare it with previous works [5], [6]. The mean is 0.13 in the considered scenario and 0.19 in the adaptation from GTA5 to Cityscapes, consistently with the value of 0.20 used in previous works [5], [6].

V. CONCLUSIONS

In this paper, we investigated the unsupervised domain adaptation task from synthetic urban scenes to real world ones. Two different strategies have been used to exploit unlabeled data. Firstly, an adversarial learning framework based on a couple of fully convolutional discriminators has been employed to align the two domain distributions. Secondly, we designed a self-teaching strategy based on the assumption that unsupervised predictions labeled as highly confident by the discriminator are reliable. Thanks to class-aware and time-varying confidence thresholds our framework can adapt to different classes and different stages of learning of the semantic segmentation network. We performed several experiments on two real datasets starting from two synthetic ones, thus proving the effectiveness and robustness of the proposed approach across different scenarios. In the future, we plan to enrich the proposed framework by enforcing the alignment of the features spaces produced from the two domains and to test it in different contexts than the road scenario.

REFERENCES

- [1] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.
- [2] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *CVPR*, 2016, pp. 3234–3243.
- [3] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou34, Y.-Y. Lin, and M.-H. Yang15, “Adversarial learning for semi-supervised semantic segmentation,” in *BMVC*, 2018.

- [4] X. Liu, J. Cao, T. Fu, Z. Pan, W. Hu, K. Zhang, and J. Liu, “Semi-supervised automatic segmentation of layer and fluid region in retinal optical coherence tomography images using adversarial learning,” *IEEE Access*, vol. 7, pp. 3046–3061, 2019.
- [5] M. Biasetton, U. Michieli, G. Agresti, and P. Zanuttigh, “Unsupervised Domain Adaptation for Semantic Segmentation of Urban Scenes,” in *CVPRW*, 2019.
- [6] U. Michieli, M. Biasetton, G. Agresti, and P. Zanuttigh, “Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation,” *IEEE Transaction on Intelligent Vehicles*, 2020.
- [7] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.
- [8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017, pp. 2881–2890.
- [9] F. Yu, V. Koltun, and T. A. Funkhouser, “Dilated residual networks,” in *CVPR*, 2017, pp. 636–644.
- [10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *TPAMI*, vol. 40, pp. 834–848, 2018.
- [11] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *ECCV*, 2018, pp. 833–851.
- [12] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, 2017.
- [13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes dataset for semantic urban scene understanding,” in *CVPR*, 2016, pp. 3213–3223.
- [14] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kotschieder, “The Mapillary vistas dataset for semantic understanding of street scenes,” in *ICCV*, 2017, pp. 4990–4999.
- [15] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.
- [16] M. Long, Y. Cao, J. Wang, and M. Jordan, “Learning transferable features with deep adaptation networks,” in *ICML*, 2015, pp. 97–105.
- [17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Unsupervised domain adaptation with residual transfer networks,” in *NeurIPS*, 2016.
- [18] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *AAAI*, 2016, pp. 2058–2065.
- [19] B. Sun and K. Saenko, “Deep CORAL: correlation alignment for deep domain adaptation,” in *ECCVW*, 2016, pp. 443–450.
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NeurIPS*, 2014, pp. 2672–2680.
- [21] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *ICML*, 2015, pp. 1180–1189.
- [22] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” vol. 17, no. 1, pp. 2096–2030, 2016.
- [23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *CVPR*, 2017, pp. 7167–7176.
- [24] M. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *NeurIPS*, 2016, pp. 469–477.
- [25] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in *CVPR*, 2017, pp. 3722–3731.
- [26] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *CVPR*, 2017, pp. 2242–2251.
- [27] S. Xie, Z. Zheng, L. Chen, and C. Chen, “Learning semantic representations for unsupervised domain adaptation,” in *ICML*, 2018.
- [28] Z. Deng, Y. Luo, and J. Zhu, “Cluster alignment with a teacher for unsupervised domain adaptation,” in *ICCV*, 2019, pp. 9944–9953.
- [29] Y. Balaji, R. Chellappa, and S. Feizi, “Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation,” in *ICCV*, 2019, pp. 6500–6508.
- [30] R. Xu, G. Li, J. Yang, and L. Lin, “Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation,” in *ICCV*, 2019, pp. 1426–1435.
- [31] S. Cicek and S. Soatto, “Unsupervised domain adaptation via regularized conditional alignment,” in *ICCV*, 2019, pp. 1416–1425.
- [32] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in *ICCV*, 2019.
- [33] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017.
- [34] J. Hoffman, D. Wang, F. Yu, and T. Darrell, “FCNs in the wild: Pixel-level adversarial and constraint-based adaptation,” *arXiv preprint arXiv:1612.02649*, 2016.
- [35] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *CVPR*, 2018, pp. 7472–7481.
- [36] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, “Domain adaptation for structured output via discriminative patch representations,” in *ICCV*, 2019, pp. 1456–1465.
- [37] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, “Cycada: Cycle-consistent adversarial domain adaptation,” in *ICML*, 2018.
- [38] Z. Murez, S. Kolouri, D. J. Kriegman, R. Ramamoorthi, and K. Kim, “Image to image translation for domain adaptation,” in *CVPR*, 2018.
- [39] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, and R. Chellappa, “Learning from synthetic data: Addressing domain shift for semantic segmentation,” in *CVPR*, 2018, pp. 3752–3761.
- [40] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, “Crdoco: Pixel-level domain transfer with cross-domain consistency,” in *CVPR*, 2019.
- [41] M. Toldo, U. Michieli, G. Agresti, and P. Zanuttigh, “Unsupervised domain adaptation for mobile semantic segmentation based on cycle consistency and feature alignment,” *Image and Vision Computing*, 2020.
- [42] F. Pizzati, R. d. Charette, M. Zaccaria, and P. Cerri, “Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation,” in *WACV*, 2020, pp. 2990–2998.
- [43] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, “Adversarial dropout regularization,” in *ICLR*, 2018.
- [44] S. Park, J. Park, S. Shin, and I. Moon, “Adversarial dropout for supervised and semi-supervised learning,” in *AAAI*, 2018.
- [45] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, “Drop to adapt: Learning discriminative features for unsupervised domain adaptation,” in *ICCV*, 2019, pp. 91–100.
- [46] M. Chen, H. Xue, and D. Cai, “Domain adaptation for semantic segmentation with maximum squares loss,” in *ICCV*, 2019.
- [47] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” in *CVPR*, 2019, pp. 2517–2526.
- [48] A. Dundar, M. Liu, T. Wang, J. Zedlewski, and J. Kautz, “Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation,” *arXiv preprint arXiv:1807.09384*, 2018.
- [49] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, and L. S. Davis, “ACE: adapting to changing environments for semantic segmentation,” *arXiv preprint arXiv:1904.06268*, 2019.
- [50] J. Choi, T. Kim, and C. Kim, “Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation,” in *ICCV*, 2019, pp. 6830–6840.
- [51] Y. Zhang, P. David, and B. Gong, “Curriculum domain adaptation for semantic segmentation of urban scenes,” in *ICCV*, 2017, pp. 2020–2030.
- [52] Q. Lian, F. Lv, L. Duan, and B. Gong, “Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach,” in *ICCV*, 2019, pp. 6758–6767.
- [53] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in *Actes de CAP 05, Conférence francophone sur l'apprentissage automatique*, 2005, pp. 281–296.
- [54] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *ECCV*, 2018, pp. 289–305.
- [55] Y. Zou, Z. Yu, X. Liu, B. V. Kumar, and J. Wang, “Confidence regularized self-training,” in *ICCV*, 2019, pp. 5982–5991.
- [56] J. Kim and C. Park, “Attribute dissection of urban road scenes for efficient dataset integration,” in *International Joint Conference on Artificial Intelligence Workshops*, 2018, pp. 8–15.
- [57] V. Nekrasov, “Pre-computed weights for ResNet-101,” <https://github.com/DrSleep/tensorflow-deeplab-resnet>, Accessed: 2020-03.
- [58] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*. Springer, 2014, pp. 740–755.
- [59] Y. Chen, W. Li, and L. Van Gool, “Road: Reality oriented adaptation for semantic segmentation of urban scenes,” in *CVPR*, 2018, pp. 7892–7901.