Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation

Swami Sankaranarayanan[1] * Yogesh Balaji [1*] Arpit Jain [2] Ser Nam Lim [2,3] Rama Chellappa [1]

[1] UMIACS, University of Maryland, College Park, MD
[2] GE Global Research, Niskayuna, NY
[3] Avitas Systems, GE Venture, Boston MA.

Abstract

Visual Domain Adaptation is a problem of immense importance in computer vision. Previous approaches showcase the inability of even deep neural networks to learn informative representations across domain shift. This problem is more severe for tasks where acquiring hand labeled data is extremely hard and tedious. In this work, we focus on adapting the representations learned by segmentation networks across synthetic and real domains. Contrary to previous approaches that use a simple adversarial objective or superpixel information to aid the process, we propose an approach based on Generative Adversarial Networks (GANs) that brings the embeddings closer in the learned feature space. To showcase the generality and scalability of our approach, we show that we can achieve state of the art results on two challenging scenarios of synthetic to real domain adaptation. Additional exploratory experiments show that our approach: (1) generalizes to unseen domains and (2) results in improved alignment of source and target distributions.

1. Introduction

Deep Convolutional Neural Networks (DCNNs) have revolutionalized the field of computer vision, achieving the best performance in a multitude of tasks such as image classification [12], semantic segmentation [20], visual question answering [23] etc. This strong performance can be attributed to the availability of abundant labeled training data. While annotating data is relatively easier for certain tasks like image classification, they can be extremely laborious and time-consuming for others. Semantic segmentation is one such task that requires great human effort as it involves obtaining dense pixel-level labels. The annotation time for obtaining pixel-wise labels for a single image from the CITYSCAPES dataset is about 1 hr., highlighting the level of difficulty ([4], [26]). The other challenge lies in collecting the data: While natural images are easier to obtain, there are certain domains like medical imaging where collecting data and finding experts to precisely label them can also be very expensive.

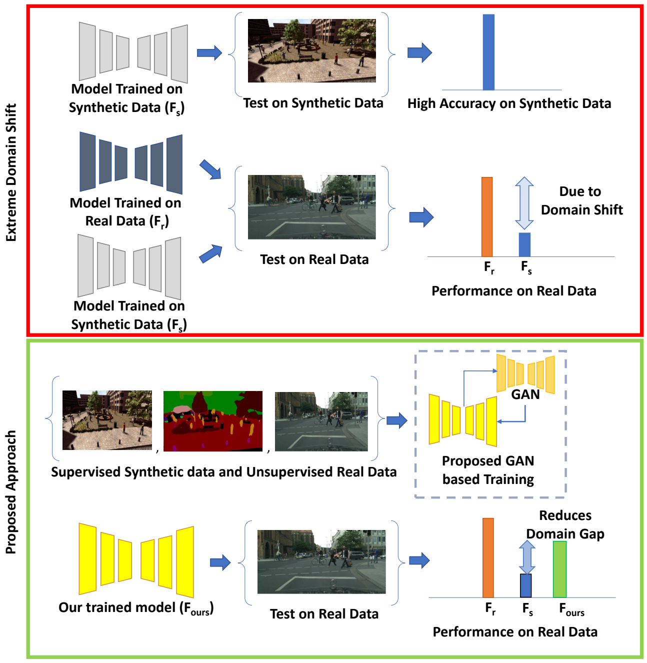

Extreme Domain Shift
Model Trained on Synthetic Data (F_s)
Test on Synthetic Data
High Accuracy on Synthetic Data
Model Trained on Real Data (F_r)
Test on Real Data
Due to Domain Shift
F_r
F_s
Performance on Real Data
Model Trained on Synthetic Data (F_s)
Proposed Approach
GAN
Supervised Synthetic data and Unsupervised Real Data
Proposed GAN based Training
Reduces Domain Gap
Our trained model (F_{ours})
Test on Real Data
F_r
F_s
F_{ours}
Performance on Real Data

Figure 1: Characterization of Domain Shift and effect of the proposed approach in reducing the same

One promising approach that addresses the above issues is the utility of synthetically generated data for training. However, models trained on the synthetic data fail to perform well on real datasets owing to the presence of domain gap between the datasets. Domain adaptation encompasses the class of techniques that address this domain shift prob-

[*] First two authors contributed equally

lem. Hence, the focus of this paper is in developing domain adaptation algorithms for semantic segmentation. Specifically, we focus on the hard case of the problem where no labels from the target domain are available. This class of techniques is commonly referred to as Unsupervised Domain Adaptation.

Traditional approaches for domain adaptation involve minimizing some measure of distance between the source and the target distributions. Two commonly used measures are Maximum Mean Discrepancy (MMD) ([9], [21] [22]), and learning the distance metric using DCNNs as done in Adversarial approaches ([7], [30]). Both approaches have had good success in the classification problems; however, as pointed out in [32], their performance improvement does not translate well to the semantic segmentation problem. This motivates the need for developing new domain adaptation techniques tailored to semantic segmentation.

The method we present in this work falls in the category of aligning domains using an adversarial framework. Among the recent techniques that address this problem, FCN in the wild [14] is the only approach that uses an adversarial framework. However, unlike [14] where a discriminator operates directly on the feature space, we project the features to the image space using a generator and the discriminator operates on this projected image space. Adversarial losses are then derived from the discriminator. We observed that applying adversarial losses in this projected image space achieved a significant performance improvement as compared to applying such losses directly in the feature space (ref. Table 4).

The main contribution of this work is that we propose a technique that employs generative models to align the source and target distributions in the feature space. We first project the intermediate feature representations obtained using a DCNN to the image space by training a reconstruction module using a combination of L_1 and adversarial losses. We then impose the domain alignment constraint by forcing the network to learn features such that source features produce target-like images when passed to the reconstruction module and vice versa. This is accomplished by employing a series of adversarial losses. As training progresses, the generation quality gradually improves, while at the same time, the features become more domain invariant.

2. Related Work

Fully Convolutional Networks (FCN) by Shelhamer *et al* [20] signified a paradigm shift in how to fully exploit the representational power of CNNs for the semantic pixel labeling tasks. While performance has been steadily improving for popular benchmarks such as PASCAL VOC [6] and MS-COCO [18], they do not address the challenges of domain shift within the context of semantic segmentation.

Domain adaptation has been widely explored in computer vision primarily for the classification task. Some of the earlier approaches involved using feature reweighting techniques [5], or constructing intermediate representations using manifolds ([11], [10]). Since the advent of deep neural networks, emphasis has been shifted to learning domain invariant features in an end-to-end fashion. A standard framework for deep domain adaptation involves minimizing a measure of domain discrepancy along with the task being solved. Some approaches use Maximum Mean Discrepancy and its kernel variants for this task ([21], [22]), while others use adversarial approaches ([7], [2], [28]).

We focus on adversarial approaches since they are more related to our work. Revgrad [7] performs domain adaptation by applying adversarial losses in the feature space, while PixelDA [2] and CoGAN [19] operate in the pixel space. While these techniques perform adaptation for the classification task, there are very few approaches aimed at semantic segmentation. To the best of our knowledge, [14] and [32] are the only two approaches that address this problem. FCN in the wild [14] proposes two alignment strategies - (1) global alignment which is an extension to the domain adversarial training proposed by [7] to the segmentation problem and (2) local alignment which aligns class specific statistics by formulating it as a multiple instance learning problem. Curriculum domain adaptation [32] on the other hand proposes curriculum-style learning approach where the easy task of estimating global label distributions over images and local distributions over landmark superpixels is learnt first. The segmentation network is then trained so that the target label distribution follow these inferred label properties.

One possible direction to address the domain adaptation problem is to employ style transfer or cross domain mapping networks to stylize the source domain images as target and train the segmentation models in this stylized space. Hence, we discuss some recent work related to the style transfer and unpaired image translation tasks. The popular work of Gatys *et al.* [8] introduced an optimization scheme involving backpropagation for performing content preserving style transfer, while Johnson *et al.* [15] proposed a feed-forward method for the same. CycleGAN [33] performs unpaired image-to-image translation by employing adversarial losses and cycle consistency losses. In our experiments, we compare our approach to some of these style-transfer based data augmentation schemes.

3. Method

In this section, we provide a formal treatment of the proposed approach and explain in detail our iterative optimization procedure. Let $X \in \mathbb{R}^{M \times N \times C}$ be an arbitrary input image (with C channels) and $Y \in \mathbb{R}^{M \times N}$ be the corresponding label map. Given an input X, we denote the output of a CNN as $\hat{Y} \in \mathbb{R}^{M \times N \times N_c}$, where N_c is the number

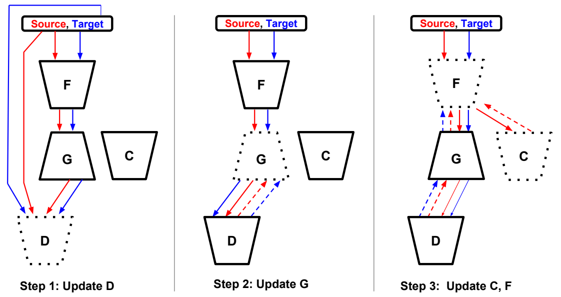


Figure 2: The directions of data flow *solid arrows* during the forward pass and gradient flow *dotted arrows* during the backward pass of our iterative update procedure. *Solid* blocks indicate that the block is frozen during that update step while *dotted* block indicate that it is being updated. Red denoted source information and Blue denotes target information.

of classes. $\hat{Y}(i, j) \in \mathbb{R}^{N_c}$ is a vector representing the class probability distribution at pixel location (i, j) output by the CNN. The source(s) or target (t) domains are denoted by a superscript such as X^s or X^t.

First, we provide an input-output description of the different network blocks in our pipeline. Next, we describe separately the treatment of source and target data, followed by a description of the different loss functions and the corresponding update steps. Finally, we motivate the design choices involved in the discriminator (D) architecture.

3.1. Description of network blocks

Our training procedure involves alternatively optimizing the following network blocks:

(a) The base network, whose architecture is similar to a pre-trained model such as VGG-16, is split into two parts: the embedding denoted by F and the pixel-wise classifier denoted by C. The output of C is a label map up-sampled to the same size as the input of F.

(b) The generator network (G) takes as input the learned embedding and reconstructs the RGB image.

(c) The discriminator network (D) performs two different tasks given an input: (a) It classifies the input as real or fake in a domain consistent manner (b) It performs a pixel-wise labeling task similar to the C network. Note that (b) is active only for source data since target data does not have any labels during training.

3.2. Treatment of source and target data

Given a source image and label pair $\{X^s, Y^s\}$ as input, we begin by extracting a feature representation using the F network. The classifier C takes the embedding $F(X^s)$ as input and produces an image-sized label map $\hat{Y}^s$. The generator G reconstructs the source input X^s conditioned on the embedding. Following recent successful works on image generation, we do not explicitly concatenate the generator input with a random noise vector but instead use dropout layers throughout the G network. As shown in Figure 3, D performs two tasks: (1) Distinguishing the real source input and generated source image as source-real/source-fake (2) producing a pixel-wise label map of the generated source image.

Given a target input X^t, the generator network G takes the target embedding from F as input and reconstructs the target image. Similar to the previous case, D is trained to distinguish between real target data (target-real) and the generated target images from G (target-fake). However, different from the previous case, D performs only a single task i.e. it classifies the target input as target-real/target-fake. Since the target data does not have any labels during training, the classifier network C is not active when the system is presented with target inputs.

3.3. Iterative optimization

Fig. 3 shows various losses used in our method. We begin by describing these losses, and then describe our iterative optimization approach.

The different adversarial losses used to train our models are shown in Table. 1. In addition to these adversarial losses, we use the following losses: (1) $\mathcal{L}_{seg}$ and $\mathcal{L}_{aux}$ - pixel-wise cross entropy loss used in standard segmentation networks such as in FCN and (2) $\mathcal{L}_{rec}$ - L_1 loss between input and reconstructed images.

The directions of flow of information across different network blocks are listed in Figure 2. In each iteration, a randomly sampled triplet (X^s, Y^s, X^t) is provided to the system. Then, the network blocks are updated iteratively in the following order:

(1) D-update: For source inputs, D is updated using a combination of within-domain adversarial loss $\mathcal{L}^s_{adv,D}$ and auxiliary classification loss $\mathcal{L}^s_{aux}$. For target inputs, it is updated using only the adversarial loss $\mathcal{L}^t_{adv,D}$. The overall loss $\mathcal{L}_D$ is given by $\mathcal{L}_D = \mathcal{L}^s_{adv,D} + \mathcal{L}^t_{adv,D} + \mathcal{L}^s_{aux}$.

(2) G-update: In this step, the generator is updated using a combination of an adversarial loss $\mathcal{L}^s_{adv,G} + \mathcal{L}^t_{adv,G}$ intended to fool D and a reconstruction loss $\mathcal{L}_{rec}$. The adversarial loss encourages realistic output from the generator. The pixelwise L_1 loss is crucial to ensure image fidelity between the generator outputs and the corresponding input images. The overall generator loss is given as: $\mathcal{L}_G = \mathcal{L}^s_{adv,G} + \mathcal{L}^t_{adv,G} + \mathcal{L}^s_{rec} + \mathcal{L}^t_{rec}$.

(3) F-update: The update to the F network is the critical aspect of our framework where the notion of domain

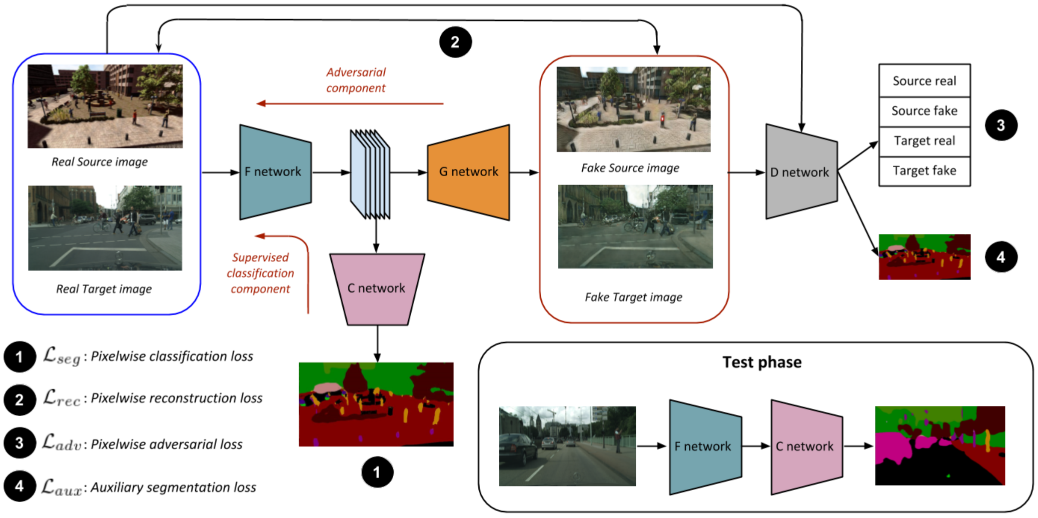

2
Adversarial component
Real Source image
Real Target image
F network
G network
Fake Source image
Fake Target image
D network
Source real
Source fake
Target real
Target fake
3
4
Supervised classification component
C network
1
$\mathcal{L}_{seg}$: Pixelwise classification loss
2
$\mathcal{L}_{rec}$: Pixelwise reconstruction loss
3
$\mathcal{L}_{adv}$: Pixelwise adversarial loss
4
$\mathcal{L}_{aux}$: Auxiliary segmentation loss
1
Test phase
F network
C network

Figure 3: During training, the F and C networks are trained jointly with the adversarial framework(G-D pair). F is updated using a combination of supervised loss and an adversarial component. In the bottom right, we show the test time usage. Only the F and C network blocks are used. There is no additional overhead during evaluation compared to the base model.

Type	Variants	Description
Within-domain	$\mathcal{L}^{s}_{adv,D}$	Classify real source input as *src-real*; fake source input as *src-fake*
	$\mathcal{L}^{s}_{adv,G}$	Classify fake source input as *src-real*
	$\mathcal{L}^{t}_{adv,D}$	Classify real target input as *tgt-real*; fake target input as *tgt-fake*
	$\mathcal{L}^{t}_{adv,G}$	Classify fake target input as *tgt-real*
Cross-domain	$\mathcal{L}^{s}_{adv,F}$	Classify fake source input as real target (*tgt-real*)
	$\mathcal{L}^{t}_{adv,F}$	Classify fake target input as real source (*src-real*)

Table 1: Within-domain and Cross-domain adversarial losses that are used to update our networks during training. G and D networks are updated using only the within-domain losses while F is updated only using the cross domain loss. All these adversarial losses originate from the D network. $\mathcal{L}_{adv,X}$ implies that the gradients from the loss function L are used to update X only, while the other networks are held fixed.

shift is captured. The parameters of F are updated using a combination of several loss terms: $\mathcal{L}_F = \mathcal{L}_{seg} + \alpha\,\mathcal{L}^{s}_{aux} + \beta\,(\mathcal{L}^{s}_{adv,F} + \mathcal{L}^{t}_{adv,F})$. As illustrated in Table 1, the adversarial loss terms used to update F account for the domain adaptation. More specifically, the iterative updates described here can be considered as a min-max game between the F and the G-D networks. During the D update step discussed earlier, the adversarial loss branch of D learns to classify the input images as real or fake in a domain consistent manner. To update F, we use the gradients from D that lead to a reversal in domain classification, i.e. for source embeddings, we use gradients from D corresponding to classifying those embeddings as from target domain ($\mathcal{L}^{s}_{adv,F}$) and for target embeddings, we use gradients from D corresponding to classifying those embeddings as from source domain ($\mathcal{L}^{t}_{adv,F}$). Note that, this is similar to the min-max game between the G-D pair, except in this case, the competition is between classifying the generated image as from source/target domains instead of them being real/fake.

3.4. Motivating design choice of D

- In traditional GANs that are derived from the DCGAN [25] implementations, the output of the discriminator is a single scalar indicating the probability of the input being fake or drawn from an underlying data distribution. Recent works on image generation have utilized the idea of *Patch* discriminator in which the output is a two dimensional feature map where each pixel carries a real/fake probability. This results in significant improvement in the visual quality of their generator reconstructions. We extend this idea to our setting by using a variant of the *Patch* discriminator, where each pixel in the output map indicates real/fake probabilities across source and target domains hence resulting in four classes per pixel: *src-real*, *src-fake*, *tgt-real*, *tgt-fake*.

- In general, GANs are hard to train on tasks which involve realistic images of a larger scale. One promising approach to training stable generative models with the GAN framework is the Auxiliary Classifier GAN (AC-GAN) approach by Odena *et al.* where they show that by conditioning G during training and adding an auxiliary classification loss to D, they can realize a more stable GAN training and even generate large scale images. Inspired by their results on image classification, we extend their idea to the segmentation problem by employing an auxiliary pixel-wise labeling loss to the D network.

Both these components prove crucial to our performance. The ablation study performed in Section 5.3 shows the effect of the above design choices on the final performance. Specific details about the architectures of these network blocks can be found in the supplementary material.

4. Experiments and Results

In this section, we provide a quantitative evaluation of our method by performing experiments on benchmark datasets. We consider two challenging synthetic datasets available for semantic segmentation: SYNTHIA and GTA-5. SYNTHIA [27] is a large dataset of photo-realistic frames rendered from a virtual city with precise pixel-level semantic annotations. Following previous works ([14], [32]), we use the SYNTHIA-RAND-CITYSCAPES subset that contains 9400 images with annotations that are compatible with cityscapes. GTA-5 is another large-scale dataset containing 24966 labeled images. The dataset was curated by Richter *et al.* [26] and is generated by extracting frames from the computer game Grand Theft Auto V.

We used CITYSCAPES [4] as our real dataset. This dataset contains urban street images collected from a moving vehicle captured in 50 cities around Germany and neighboring countries. The dataset comes with 5000 annotated images split into three sets - 2975 images in the *train* set, 500 images in the *val* set and 1595 images in the *test* set. In all our experiments, for training our models we used labeled SYNTHIA or GTA-5 dataset as our source domain and unlabeled CITYSCAPES *train* set as our target domain. We compared the proposed approach with the only two contemporary methods that address this problem: FCN in the wild [14] and Curriculum Domain adaptation [32]. Following these approaches, we designate the 500 images from CITYSCAPES *val* as our test set.

Architecture In all our experiments, we used FCN-8s as our base network. The weights of this network were initialized with the weights of the VGG-16 [29] model trained on Imagenet [17].

Implementation details In all our experiments, images were resized and cropped to 1024×512. We trained our model for $100,000$ iterations using Adam solver [16] with a batch size of 1. Learning rate of 10^{-5} was used for F and C networks, and 2×10^{-4} for G and D networks. While evaluating on CITYSCAPES dataset whose images and ground truth annotations are of size 2048×1024, we first produce our predictions on the 1024×512 sized image and then upsample our predictions by a factor of 2 to get the final label map, which is used for evaluation. Our training codes and additional results are publicly available. [1]

4.1. SYNTHIA -> CITYSCAPES

In this experiment, we use the SYNTHIA dataset as our source domain, and CITYSCAPES as our target domain. We randomly pick 100 images from the 9400 labeled images of SYNTHIA dataset and use it for validation purposes, the rest of the images are used for training. We use the unlabeled images corresponding to the CITYSCAPES *train* set for training our model. In order to ensure fairness of experimental results, we followed the exact evaluation protocol as specified by the previous works ([14],[32]): The 16 common classes between SYNTHIA and CITYSCAPES are chosen used as our labels. The predictions corresponding to the other classes are treated as belonging to void class, and not backpropagated during training. The 16 classes are: sky, building, road, sidewalk, fence, vegetation, pole, car, traffic sign, person, bicycle, motorcycle, traffic light, bus, wall, and rider.

Table 2a reports the performance of our method in comparison with [14] and [32]. The source-only model which corresponds to the no adaptation case i.e. training only using the source domain data achieves a mean IOU of 26.8. The target-only values denote the performance obtained by a model trained using CITYSCAPES *train* set (supervised training), and they serve as a crude upper bound to the domain adaptation performance. These values were included to put in perspective the performance gains obtained by the proposed approach. We observe that our method achieves a mean IOU of 36.1, thereby improving the baseline by **9.3 points**, thus resulting in a higher performance improvement compared to other reported methods.

4.2. GTA5 -> CITYSCAPES

In this experiment, we adapt from the GTA-5 dataset to the CITYSAPES dataset. We randomly pick 1000 images from the 24966 labeled images of GTA-5 dataset and use it for validation purpose and use the rest of the images for

[1] Training code: https://goo.gl/3Jsu2s

Method	Base n/w	road	sidewalk	bldg	wall	fence	pole	t light	t sign	veg	sky	person	rider	car	bus	mbike	bike	mIOU	mIOU gain
Source only [14]	Dilation-Frontend	6.4	17.7	29.7	1.2	0.0	15.1	0.0	7.2	30.3	66.8	51.1	1.5	47.3	3.9	0.1	0.0	17.4	
FCN wild [14]	[31]	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	20.2	2.8
Source only [32]	FCN8s-VGG16	5.6	11.2	59.6	8.0	0.5	21.5	8.0	5.3	72.4	75.6	35.1	9.0	23.6	4.5	0.5	18.0	22.0	
Curr. DA [32]	[20]	65.2	26.1	74.9	0.1	0.5	10.7	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	29.0	7.0
Ours - Source only	FCN8s-VGG16	30.1	17.5	70.2	5.9	0.1	16.7	9.1	12.6	74.5	76.3	43.9	13.2	35.7	14.3	3.7	5.6	26.8	
Ours - Adapted	[20]	80.1	29.1	77.5	2.8	0.4	26.8	11.1	18.0	78.1	76.7	48.2	15.2	70.5	17.4	8.7	16.7	**36.1**	**9.3**
Target-only	FCN8s-VGG16	96.5	74.6	86.1	37.1	33.2	30.2	39.7	51.6	87.3	90.4	60.1	31.7	88.4	52.3	33.6	59.1	59.5	-

(a) SYNTHIA → CITYSCAPES

Method	Base n/w	road	sidewalk	bldg	wall	fence	pole	t light	t sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIOU	mIOU gain
Source only [14]	Dilation-Frontend	31.9	18.9	47.7	7.4	3.1	16.0	10.4	1.0	76.5	13.0	58.9	36.0	1.0	67.1	9.5	3.7	0.0	0.0	0.0	21.2	
FCN wild [14]	[31]	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	5.9
Source only [32]	FCN8s-VGG16	18.1	6.8	64.1	7.3	8.7	21.0	14.9	16.8	45.9	2.4	64.4	41.6	17.5	55.3	8.4	5.0	6.9	4.3	13.8	22.3	
Curr. DA [32]	[20]	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	16.6	28.9	6.6
Ours - Source only	FCN8s-VGG16	73.5	21.3	72.3	18.9	14.3	12.5	15.1	5.3	77.2	17.4	64.3	43.7	12.8	75.4	24.8	7.8	0.0	4.9	1.8	29.6	
Ours - Adapted	[20]	88.0	30.5	78.6	25.2	23.5	16.7	23.5	11.6	78.7	27.2	71.9	51.3	19.5	80.4	19.8	18.3	0.9	20.8	18.4	**37.1**	**7.5**
Target-only	FCN8s-VGG16	96.5	74.6	86.1	37.1	33.2	30.2	39.7	51.6	87.3	52.6	90.4	60.1	31.7	88.4	54.9	52.3	34.7	33.6	59.1	57.6	-

(b) GTA5 → CITYSCAPES

Table 2: Results of Semantic Segmentation by adapting from (a) SYTNHIA to CITYSCAPES and (b) GTA-5 to CITYSCAPES. We compare with two approaches that use two different base networks. To obtain a fair idea about our performance gain, we compare with the Curriculum DA approach that uses the same base network as ours. The Target-only training procedure is the same for both the settings since in both cases the target domain is CITYSCAPES. However, the results in (a) are reported over the 16 common classes while the results in (b) are reported over all the 19 classes.

training. We use the unlabeled images corresponding to the CITYSCAPES *train* set for training our model. In order to ensure fairness of experimental results, we followed the exact evaluation protocol as specified by the previous works ([14], [32]): we use 19 common classes between GTA-5 and CITYSCAPES as our labels. The results of this experiment are reported in Table. 2b. Similar to the previous experiment, our baseline performance (29.6) is higher than the performance reported in [14], due to difference in network architecture and experimental settings. On top of this, the proposed approach yields an improvement of 7.5 points to obtain a mIOU of **37.1**. This performance gain is higher than that achieved by the other compared approaches.

Note regarding different baselines: The baseline numbers reported by us do not match with the ones reported in [32] and [14] due to different experimental settings (this mismatch was also reported in [32]). However, we would like to point out that we improve over a stronger baseline compared to the other two methods in both our adaptation experiments. In addition, [32] uses additional data from PASCAL-CONTEXT [24] dataset to obtain the superpixel segmentation. In contrast, our approach is a single stage end-to-end learning framework that does not use any additional data and yet obtains better performance improvement.

5. Discussion

In this section, we perform several exploratory studies to give more insight into the functionality and effectiveness of the proposed approach. similar to the previous section, all the evaluation results are reported on the CITYSCAPES *val* set, unless specified otherwise. We denote this set as the *test set*.

5.1. Effect of Image Size

The datasets considered in this paper consists of images of large resolution which is atleast twice larger than the most commonly used Segmentation benchmarks for CNNs i.e. PASCAL VOC (500×300) and MSCOCO (640×480). In this setting, it is instructive to understand the effect of image size on the performance of our algorithm both from a quantitative and computational perspective. Table 3 presents the results of our approach applied over three different image sizes along with the training and evaluation times. It should be noted that the Curriculum DA approach [32] used a resolution of 640×320. By comparing with our main results in Table 2a, we see that our approach provides a higher relative performance improvement over a similar baseline.

Table 3: Mean IoU values and computation times across different image size on the SYNTHIA → CITYSCAPES setting. The numbers in **bold** indicate the absolute improvement in performance over the *Source-only* baseline. The reported training and evaluation times are for the proposed approach and are averaged over training and evaluation runs.

Image size	512×256	640×320	1024×512
mIOU-*Source-only*	21.5	23.2	26.8
mIOU-*Ours*	31.3 (**+9.8**)	34.5 (**+11.3**)	36.1 (**+9.3**)
Train time (per image)	1.5s	2.1s	2.9s
Eval time (per image)	0.16s	0.19s	0.3s

5.2. Comparison with direct style transfer

Generative methods for style transfer have achieved a great amount of success in the recent past. A simple approach to performing domain adaptation is to use such approaches as a data augmentation method: transfer the images from the source domain to target domain and use the provided source ground truth to train a classifier on the combined source and target data. In order to compare the proposed approach with this direct data augmentation procedure, we used a state of the art generative approach (CycleGAN [33]) to transfer images from source domain to target domain. From our experiment, using generative approaches solely as a data augmentation method provides only a relatively small improvement over the source-only baseline and clearly suboptimal compared to the proposed approach. However, as shown in a recent approach by Hoffman *et al.* [13], such cross domain transfer can be performed by a careful training procedure. The results obtained by the proposed approach is comparable or better then [13]. Combining both approaches to produce a much stronger domain adaptation technique for segmentation is under progress.

5.3. Component-wise ablation

In this experiment, we show how each component in our loss function affects the final performance. We consider the following cases: (a) Ours(full): the full implementation of our approach (b) Ours w/o auxiliary pixel-wise loss: Here, the output of the D network is a single branch classifying the input as real/fake. This corresponds to $\alpha = 0$ in the F-update step. Note that, setting both α and β as zero corresponds to the source-only setting in our experiments. Setting only $\beta = 0$ does not improve over the source-only baseline as there is no cross domain adversarial loss. (c) Ours w/o *Patch* discriminator: Instead of using the D network as a *Patch* discriminator, we used a regular GAN-like discriminator where the output is a 4-D probability vector that the input image belongs to one of the four classes - *src-real*, *src-fake*, *tgt-real* and *tgt-fake*. (d) Feature space based D: In this setting, we remove the G-D networks and apply an adversarial loss directly on the embedding. This is similar to the global alignment setting in the FCN-in-the-wild approach [14].

The mean IoU results on the *test set* are shown in Table. 4. It can be observed that each component is very important to obtain the full improvement in performance.

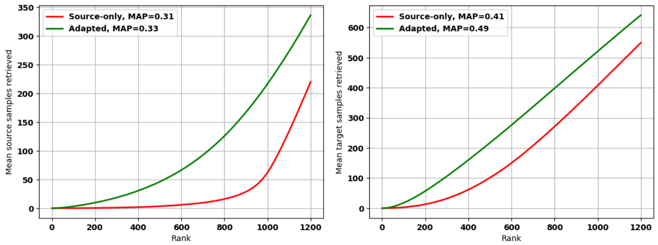
Table 4: Ablation study showing the effect of each component on the final performance of our approach on the SYNTHIA → CITYSCAPES setting

Method	mean IoU
Source-only	22.2
Feature space based D	25.3
Ours w/o *Patch* Discriminator	28.3
Ours w/o auxiliary loss ($\alpha = 0$)	29.2
Ours	**34.5**

5.4. Cross Domain Retrieval

A crucial aspect of domain adaptation is in finding good measures of domain discrepancy that provide a good illustration of the domain shift. While there exist several classical measures such as $\mathcal{A}$-distance [1] and MMD [9] for the case of image classification, the extension of such measures for a pixel-wise problem such as semantic segmentation is non-trivial. In this section, we devise a simple experiment in order to illustrate how the proposed approach brings source and target distributions closer in the learnt embedding space. We start with the last layer of the F network, which we label as the embedding layer, whose output is a spatial feature map. We perform an average pooling to reduce this spatial map to a 4096 dimensional feature descriptor for each input image.

We begin the cross domain retrieval task by choosing a pool of $N = N_{src} + N_{tgt}$ images from the combined source and target training set. Let X denote these set of images and F_X denote the set of the feature descriptors computed for X. Then, we choose two query sets, one consisting of source images (S) and the other consisting of target images (T), each disjoint with X. Let the corresponding feature sets be denoted as Q_S and Q_T. We retrieve k-NN lists for each item in the query set from the combined feature set F_X. For each query point in Q_S, we count the number of target samples retrieved in the corresponding k-NN list. $|A_k|$ indicates the average number of target samples retrieved over the entire source query set Q_S. For each query point in Q_T, we count the number of source samples retrieved in the corresponding k-NN list. $|B_k|$ indicates the average number of source samples retrieved over the entire target query set Q_T. We used cosine similarity as a metric to compute the k-NN lists. If more target samples are retrieved for a source query point (and vice-versa), it suggests that source and target distributions are aligned well in the feature space.



(a) Target → Source, $|B_k|$ (vs) k (b) Source → Target, $|A_k|$ (vs) k

Figure 4: Illustration of Domain Adaptation achieved by the proposed approach. The plot compares the average number of retrieved sampled for the cross domain retrieval task described in Section 5.4 between the source-only model and the model adapted using the proposed approach. Target → Source implies that the query set used belongs to target domain (Q_T) and items queried for from the set X belong to the source domain and vice-versa for Source → Target. In general, the values plotted on the y-axis corresponds to the number of samples retrieved from the set X that belong to the opposite domain as to that of the query set.

For this experiment, the sizes of query sets and the feature set F_X are as follows: N_{src} = N_{tgt} = 1000, $|Q_S|$ = 1000, $|Q_T|$ = 1000. The mean average precision (mAP) was computed across the entire query sets for the respective cross domain tasks. Figure 4 shows the plot of the quantities $|A_k|$ (Fig.4b) and $|B_k|$ (Fig.4a) for a range of values of k. It can be observed from the plots in both the tasks that for any given rank k, the number of cross domain samples retrieved by the adapted model is higher than the source-only model. This effect becomes more clear as k increases. This observation is supported by better mAP values for the adapted model as shown in Figure 4. While this by itself is not a sufficient condition for better segmentation performance, however this along with the results from Table 2 imply that the proposed approach performs domain adaptation in a meaningful manner. Owing to the difficulty in visualizing the mapping learned for segmentation tasks, a cross domain retrieval experiment can be seen as a reasonable measure of how domain gap is reduced in the feature space.

5.5. Generalization to unseen domains

A desirable characteristic of any domain adaptation algorithm is domain generalization i.e. improving performance over domains that are not seen during training. To test the generalization capability of the proposed approach, we test the model trained for the SYNTHIA → CITYSCAPES setting on the CamVid dataset [3]. We choose to evaluate our models on the 10 common classes among the three datasets. Table 5 shows the mean IoU values computed for the source-only baseline and the adapted model. The proposed approach yields a raw improvement of **8.3** points in performance which is a significant improvement considering the fact that CamVid images are not seen by the adapted model during training. This experiment showcases the ability of the proposed approach to learn domain invariant representations in a generalized manner.

Table 5: Mean IoU segmentation performance measured on a third unseen domain (CamVid dataset) for the models corresponding to the SYNTHIA → CITYSCAPES setting

Method	mean IoU
Source-only	36.1
Ours	**44.4**

6. Conclusion and Future Work

In this paper, we have addressed the problem of performing semantic segmentation across different domains. In particular, we have considered a very hard case where abundant supervisory information is available for synthetic data (source) but no such information is available for real data (target). We proposed a joint adversarial approach that transfers the information of the target distribution to the learned embedding using a generator-discriminator pair. We have shown the superiority of our approach over existing methods that address this problem using experiments on two large scale datasets thus demonstrating the generality and scalability of our training procedure. Furthermore, our approach has no extra computational overhead during evaluation, which is a critical aspect when deploying such methods in practice. As future work, we would like to extend this approach to explicitly incorporate geometric constraints accounting for perspective variations and to adapt over temporal inputs such as videos across different domains.

7. Acknowledgement

The Authors acknowledge support of the following organisations for sponsoring this work: (1) Avitas Systems, a GE venture (2) MURI from the Army Research Office under the Grant No. W911NF-17-1-0304. This is part of the collaboration between US DOD, UK MOD and UK Engineering and Physical Research Council (EPSRC) under the Multidisciplinary University Research Initiative.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007. 7

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016. 2

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 8

[4] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[5] H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, June 2007. 2

[6] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 2010. 2

[7] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 2

[8] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 2

[9] B. Geng, D. Tao, and C. Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011. 2, 7

[10] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. 2

[11] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 2011. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[13] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 7

[14] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 2, 5, 6, 7

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 694–711, 2016. 2

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 5

[18] T.-Y. Lin et al. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[19] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 469–477. 2016. 2

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2, 6

[21] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 97–105, 2015. 2

[22] M. Long, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *CoRR*, abs/1602.04433, 2016. 2

[23] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016. 1

[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[25] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4

[26] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016. 1, 5

[27] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 5

[28] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *arXiv preprint arXiv:1704.01705*, 2017. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[30] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *CoRR*, abs/1702.05464, 2017. 2

[31] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 6

[32] Y. Zhang, P. David, and B. Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 5, 6

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 7