

# Image to Image Translation for Domain Adaptation

Zak Murez<sup>1,2</sup> Soheil Kolouri<sup>2</sup> David Kriegman<sup>1</sup> Ravi Ramamoorthi<sup>1</sup> Kyungnam Kim<sup>2</sup>  
<sup>1</sup> University of California, San Diego; <sup>2</sup> HRL Laboratories, LLC;  
 {zmurez, kriegman, ravir}@cs.ucsd.edu, {skolouri, kkim}@hrl.com

## Abstract

We propose a general framework for unsupervised domain adaptation, which allows deep neural networks trained on a source domain to be tested on a different target domain without requiring any training annotations in the target domain. This is achieved by adding extra networks and losses that help regularize the features extracted by the backbone encoder network. To this end we propose the novel use of the recently proposed unpaired image-to-image translation framework to constrain the features extracted by the encoder network. Specifically, we require that the features extracted are able to reconstruct the images in both domains. In addition we require that the distribution of features extracted from images in the two domains are indistinguishable. Many recent works can be seen as specific cases of our general framework. We apply our method for domain adaptation between MNIST, USPS, and SVHN datasets, and Amazon, Webcam and DSLR Office datasets in classification tasks, and also between GTA5 and Cityscapes datasets for a segmentation task. We demonstrate state of the art performance on each of these datasets.

## 1. Introduction

The recent unprecedented advances in computer vision and machine learning are mainly due to: 1) deep (convolutional) neural architectures, and 2) existence of abundant labeled data. Deep convolutional neural networks (CNNs) [17, 10, 13] trained on large numbers of labeled images (tens of thousands to millions) provide powerful image representations that can be used for a wide variety of tasks including recognition, detection, and segmentation. On the other hand, obtaining abundant annotated data remains a cumbersome and expensive process in the majority of applications. Hence, there is a need for transferring the learned knowledge from a source domain with abundant labeled data to a target domain where data is unlabeled or sparsely labeled. The major challenge for such knowledge transfer is a phenomenon known as domain shift [8], which refers to the different distribution of data in the target domain com-

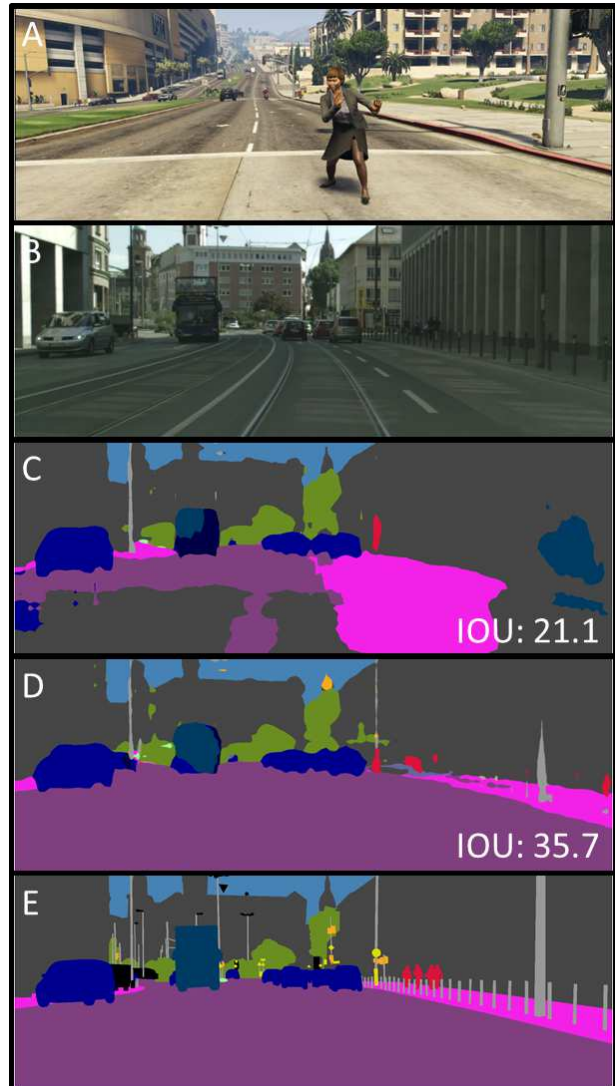


Figure 1. A) Sample image from the synthetic GTA5 dataset. B) Input image from the real Cityscapes dataset. C) Segmentation result trained on GTA5 dataset without any domain adaptation. D) Ours. E) Ground truth. We can see that our adaptation fixes large areas of simple mistakes on the road and sidewalk and building on the right. We also partially detect the thin pole on the right. The mean Intersection Over Union (IOU) values are reported.

pared to the source domain.

To further motivate the problem, consider the emerging application of autonomous driving where a semantic segmentation network is required to be trained to detect roads, cars, pedestrians, etc. Training such segmentation networks requires semantic, instance-wise, dense pixel annotations for each scene, which is excruciatingly expensive and time consuming to acquire. To avoid human annotations, a large body of work focuses on designing photo-realistic simulated scenarios in which the ground truth annotations are readily available. Synthia [27], Virtual KITTI [3], and GTA5 [26] datasets are examples of such simulations, which include a large number of synthetically generated driving scenes together with ground truth pixel-level semantic annotations. Training a CNN based on such synthetic data and applying it to real-world images (i.e. from a dashboard mounted camera), such as the Cityscapes dataset [2], will give very poor performance due to the large differences in image characteristics which gives rise to the domain shift problem. Figure 1 demonstrates this scenario where a network is trained on the GTA5 dataset [26], which is a synthetic dataset, for semantic segmentation and is tested on the Cityscapes dataset [2]. It can be seen that with no adaptation the network struggles with segmentation (Figure 1, C), while our proposed framework ameliorates the domain shift problem and provides a more accurate semantic segmentation.

Domain adaptation techniques aim to address the domain shift problem, by finding a mapping from the source data distribution to the target distribution. Alternatively, both domains could be mapped into a shared domain where the distributions are aligned. Generally, such mappings are not unique and there exist many mappings that align the source and target distributions. Therefore various constraints are needed to narrow down the space of feasible mappings. Recent domain adaptation techniques parameterize and learn these mappings via deep neural networks [34, 21, 35, 23, 30, 32]. In this paper, we propose a unifying, generic, and systematic framework for unsupervised domain adaptation, which is broadly applicable to many image understanding and sensing tasks where training labels are not available in the target domain. We further demonstrate that many existing methods for domain adaptation arise as special cases of our framework.

While there are significant differences between the recently developed domain adaptation methods, a common and unifying theme among these methods can be observed. We identify three main attributes needed to achieve successful unsupervised domain adaptation: 1) domain agnostic feature extraction, 2) domain specific reconstruction, and 3) cycle consistency. The first requires that the distributions of features extracted from both domains are indistinguishable (as judged by an adversarial discriminator network). This

idea was utilized in many prior methods [12, 4, 5], but alone does not give a strong enough constraint for domain adaptation knowledge transfer, as there exist many mappings that could match the source and target distributions in the shared space. The second is requiring that the features are able to be decoded back to the source and target domains. This idea was used in Ghifary et al. [6] for unsupervised domain adaptation. Finally, the cycle consistency is needed for unpaired source and target domains to ensure that the mappings are learned correctly and they are well-behaved, in the sense that they do not collapse the distributions into single modes [40]. Figure 2 provides a high-level overview of our framework.

The interplay between the ‘domain agnostic feature extraction’, ‘domain specific reconstruction with cycle consistency’, and ‘label prediction from agnostic features’ enables our framework to simultaneously learn from the source domain and adapt to the target domain. By combining all these different components into a single unified framework we build a systematic framework for domain knowledge transfer that provides an elegant theoretical explanation as well as improved experimental results. We demonstrate the superior performance of our proposed framework for segmentation adaptation from synthetic images to real world images (See Figure 1 as an example), as well as for classifier adaptation on three digit datasets. Furthermore, we show that many of the State Of the Art (SOA) methods can be viewed as special cases of our proposed framework. Code is available at <https://github.com/zmure/zmurez/I2IAdapt>.

## 2. Related Work

There has been a plethora of recent work in the field of visual domain adaptation addressing the domain shift problem [8], otherwise known as the dataset bias problem. The majority of recent work use deep convolutional architectures to map the source and target domains into a shared space where the domains are aligned [36, 34, 35, 12, 30]. These methods widely differ on the architectures as well as the choices of loss functions used for training them. Some have used Maximum Mean Discrepancy (MMD) between the distributions of the source and target domains in the shared space [21], while others have used correlation maximization to align the second-order statistics of the domains. Another popular and effective choice is maximizing the confusion rate of an adversarial network, that is required to distinguish the source and target domains in the shared space [34, 12, 4, 5, 6]. Other approaches include the work by Sener et al. [31], where the domain transfer is formulated in a transductive setting, and the Residual Transfer Learning (RTL) approach [22] where the authors assume that the source and target classifiers only differ by a residual function and learn these residual functions.

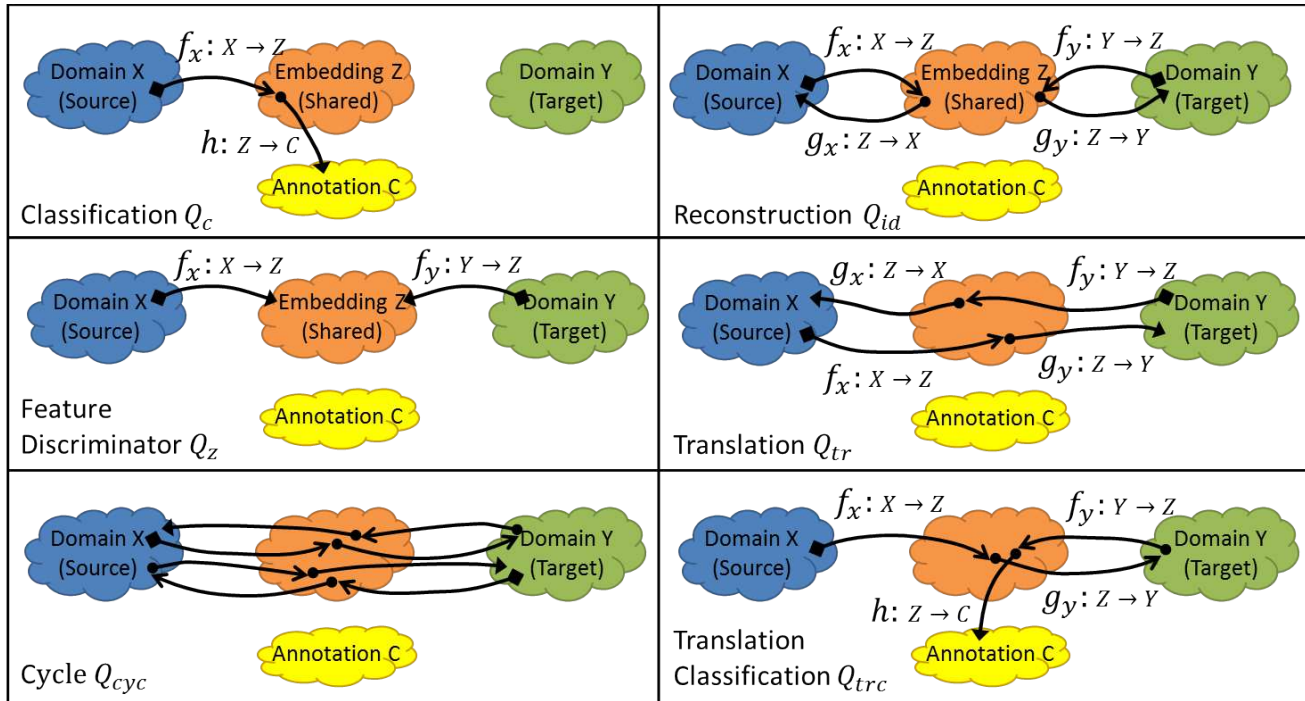


Figure 2. The detailed system architecture of our I2I (image to image) Adapt framework. The pathways to the loss modules denote the inputs to these modules, which are used for training. Best viewed in color.

Our work is primarily motivated by the work of Hoffman et al. [12], Isola et al. [15], Zhu et al. [40], and Ghifary et al. [6]. Hoffman et al. [12] utilized fully convolutional networks with domain adversarial training to obtain domain agnostic features (i.e. shared space) for the source and target domains, while constraining the shared space to be discriminative for the source domain. Hence, by learning the mappings from source and target domains to the shared space (i.e.  $f_x$  and  $f_y$  in Figure 2), and learning the mapping from the shared space to annotations (i.e.  $h$  in Figure 2), their approach effectively enables the learned classifier to be applicable to both domains. The Deep Reconstruction Classification Network (DRCN) of Ghifary et al. [6], utilizes a similar approach but with a constraint that the embedding must be decodable, and learns a mapping from the embedding space to the target domain (i.e.  $g_y$  in Figure 2). The image-to-image translation work by Isola et al. [15] maps the source domain to the target domain by an adversarial learning of  $f_x$  and  $g_y$  and composing them  $g_y \circ f_x: X \rightarrow Y$ . In their framework the target and source images were assumed to be paired, in the sense that for each source image there exists a known corresponding target image. This assumption was lifted in the follow-up work of Zhu et al. [40] and Royer et al. [28], where cycle consistency was used to learn the mappings based on unpaired source and target images. While the approaches of Isola et al. [15] and Zhu et al. [40] do not address the domain adaptation problem, yet

they provide a baseline for learning high quality mappings from a visual domain into another.

The patterns that collectively emerge from the mentioned papers [36, 12, 15, 6, 40], are: a) the shared space must be a discriminative embedding for the source domain, b) the embedding must be domain agnostic, hence maximizing the similarity between the distributions of embedded source and target images, c) the information preserved in the embedding must be sufficient for reconstructing domain specific images, d) adversarial learning as opposed to the classic losses can significantly enhance the quality of learned mappings, e) cycle-consistency is required to reduce the space of possible mappings and ensure their quality, when learning the mappings from unpaired images in the source and target domains. Our proposed method for unsupervised domain adaptation unifies the above-mentioned pieces into a generic framework that simultaneously solves the domain adaptation and image-to-image translation problems.

There have been other recent efforts toward a unifying and general framework for deep domain adaptation. The Adversarial Discriminative Domain Adaptation (ADDA) work by Tzeng et al. [35] is an instance of such frameworks. Tzeng et al. [35] identify three design choices for a deep domain adaptation system, namely a) whether to use a generative or discriminative base, whether to share mapping parameters between  $f_x$  and  $f_y$ , and the choice of adversarial training. They observed that modeling image distributions



might not be strictly necessary if the embedding is domain agnostic (i.e. domain invariant).

Very similar ideas to ours have been published concurrently by Hoffman et al. [11] and Liu et al. [19].

### 3. Method

Consider training images  $x_i \in X$  and their corresponding annotations/labels  $c_i \in C$  from the source domain (i.e. domain  $X$ ). Note that  $c_i$  may be image level such as in classification or pixel level in the case of semantic segmentation. Also consider training images  $y_j \in Y$  in the target domain (i.e. domain  $Y$ ), where we do not have corresponding annotations for these images. Our goal is then to learn a classifier that maps the target images,  $y_j$ s, to labels  $C$ . We note that the framework is readily extensible to a semi-supervised learning or few-shot learning scenario where we have annotations for a few images in the target domain. Given that the target domain lacks labels, the general approach is to learn a classifier on the source domain and adapt it in a way that its domain distribution matches that of the target domain.

The overarching idea here is to find a joint latent space,  $Z$ , for the source and target domains,  $X$  and  $Y$ , where the representations are domain agnostic. To clarify this point, consider the scenario in which  $X$  is the domain of driving scenes/images on a sunny day and  $Y$  is the domain of driving scenes on a rainy day. While ‘sunny’ and ‘rainy’ are characteristics of the source and target domains, they are truly nuisance variations with respect to the annotation/classification task (e.g. semantic segmentation of the road), as they should not affect the annotations. Treating such characteristics as structured noise, we would like to find a latent space,  $Z$ , that is invariant to such variations. In other words, domain  $Z$  should not contain domain specific characteristics, hence it should be domain agnostic. In what follows we describe the process that leads to finding such a domain agnostic latent space.

Let the mappings from source and target domains to the latent space be defined as  $f_x : X \rightarrow Z$  and  $f_y : Y \rightarrow Z$ , respectively (See Figure 2). In our framework these mappings are parameterized by deep convolutional neural networks (CNNs). Note that the members of the latent space  $z \in Z$  are high dimensional vectors in the case of image level tasks, or feature maps in the case of pixel level tasks. Also, let  $h : Z \rightarrow C$  be the classifier that maps the latent space to labels/annotations (i.e. the classifier module in Figure 2). Given that the annotations for the source class  $X$  are known, one can define a supervised loss function to enforce  $h(f_x(x_i)) = c_i$ :

$$Q_c = \sum_i l_c(h(f_x(x_i)), c_i) \quad (1)$$

where  $l_c$  is an appropriate loss (e.g. cross entropy for clas-

sification and segmentation). Minimizing the above loss function leads to the standard approach of supervised learning, which does not concern domain adaptation. While this approach would lead to a method that performs well on the images in the source domain,  $x_i \in X$ , it will more often than not perform poorly on images from the target domain  $y_j \in Y$ . The reason is that, domain  $Z$  is biased to the distribution of the structured noise (‘sunny’) in domain  $X$  and the structured noise in domain  $Y$  (‘rainy’) confuses the classifier  $h(\cdot)$ . To avoid such confusion we require the latent space,  $Z$ , to be domain agnostic, so it is not sensitive to the domain specific structured noise. To achieve such a latent space we systematically introduce a variety of auxiliary networks and losses to help regularize the latent space and consequently achieve a robust  $h(\cdot)$ . The auxiliary networks and loss pathways are depicted in Figure 2. In what follows we describe the individual components of the regularization losses.

1. First of all  $Z$  is required to preserve the core information of the target and source images and only discard the structured noise. To impose this constraint on the latent space, we first define decoders  $g_x : Z \rightarrow X$  and  $g_y : Z \rightarrow Y$  that take the features in the latent space to the source and target domains, respectively. We assume that if  $Z$  retains the crucial/core information of the domains and only discards the structured noise, then the decoders should be able to add the structured noise back and reconstruct each image from their representation in the latent feature space,  $Z$ . In other words, we require  $g_x(f_x(\cdot))$  and  $g_y(f_y(\cdot))$  to be close to identity functions/maps. This constraint leads to the following loss function:

$$Q_{id} = \sum_i l_{id}(g_x(f_x(x_i)), x_i) + \sum_j l_{id}(g_y(f_y(y_j)), y_j) \quad (2)$$

where  $l_{id}(\cdot, \cdot)$  is a pixel-wise image loss such as the  $L_1$  norm.

2. We would like the latent space  $Z$  to be domain agnostic. This means that the feature representations of the source and target domain should not contain domain specific information. To achieve this, we use an adversarial setting in which a discriminator  $d_z : Z \rightarrow \{c_x, c_y\}$  tries to classify if a feature in the latent space  $z \in Z$  was generated from domain  $X$  or  $Y$ , where  $c_x$  and  $c_y$  are binary domain labels (i.e. from domain  $X$  or domain  $Y$ ). The loss function then can be defined as the certainty of the discriminator (i.e. domain agnosticism is equivalent to fooling the discriminator), and

therefore we can formulate this as:

$$Q_z = \sum_i l_a(d_z(f_x(x_i)), c_x) + \sum_j l_a(d_z(f_y(y_j)), c_y) \quad (3)$$

where  $l_a(\cdot, \cdot)$  is an appropriate loss (the cross entropy loss in traditional GANs [7] and mean square error in least squares GAN [24]). The discriminator is trained to maximize this loss while the discriminator is trained to minimize it.

3. To further ensure that the mappings  $f_x$ ,  $f_y$ ,  $g_x$ , and  $g_y$  are consistent we define translation adversarial losses. An image from target (source) domain is first encoded to the latent space and then decoded to the source (target) domain to generate a ‘fake’ (translated) image. Next, we define discriminators  $d_x : X \rightarrow \{c_x, c_y\}$  and  $d_y : Y \rightarrow \{c_x, c_y\}$ , to identify if an image is ‘fake’ (generated from the other domain) or ‘real’ (belonged to the actual domain). To formulate this translation loss function we can write:

$$Q_{tr} = \sum_i l_a(d_y(g_y(f_x(x_i))), c_x) + \sum_j l_a(d_x(g_x(f_y(y_j))), c_y) \quad (4)$$

4. Given that there are no correspondences between the images in the source and target domains, we need to ensure that the semantically similar images in both domains are projected into close vicinity of one another in the latent space. To ensure this, we define the cycle consistency losses where the ‘fake’ images generated in the translation loss,  $g_x(f_y(y_j))$  or  $g_y(f_x(x_i))$ , are encoded back to the latent space and then decoded back to their original space. The entire cycle should be equivalent to an identity mapping. We can formulate this loss as follows:

$$Q_{cyc} = \sum_i l_{id}(g_x(f_y(g_y(f_x(x_i))))), x_i) + \sum_j l_{id}(g_y(f_x(g_x(f_y(y_j))))), y_j) \quad (5)$$

5. To further constrain the translations to maintain the same semantics, and allow the target encoder to be trained with supervision on target domain ‘like’ images we also define a classification loss between the source to target translations and the original source labels:

$$Q_{trc} = \sum_i l_c(h(f_y(g_y(f_x(x_i))))), c_i) \quad (6)$$

Method	$\lambda_c$	$\lambda_z$	$\lambda_{tr}$	$\lambda_{id_x}$	$\lambda_{id_y}$	$\lambda_{cyc}$	$\lambda_{trc}$
[12]	✓	✓					
[35]	✓	✓					
[6]	✓				✓		
[30]	✓		✓				
[40]			✓			✓	
Ours	✓	✓	✓	✓	✓	✓	✓

Table 1. Showing the relationship between the existing methods and our proposed method.

Finally, by combining these individual losses we define the general loss to be,

$$Q = \lambda_c Q_c + \lambda_z Q_z + \lambda_{tr} Q_{tr} + \lambda_{id} Q_{id} + \lambda_{cyc} Q_{cyc} + \lambda_{trc} Q_{trc} \quad (7)$$

A variety of prior methods for domain adaptation are special cases of our framework. Table 1 summarizes which hyperparameters to include and which are set to zero to recover these prior methods.

## 4. Experiments

The loss function is optimized via the ADAM method with learning rate 0.0002 and betas 0.5 and 0.999, in an end-to-end manner. The discriminative networks,  $d_x$ ,  $d_y$ , and  $d_z$  are trained in an alternating optimization alongside with the encoders and decoders.

To further constrain the features that are learned we share the weights of the encoders. We also share the weights of the first few layers of the decoders. To stabilize the discriminators we train them using the Improved Wasserstein method [9]. The loss of the feature discriminator ( $Q_z$ ) is only backpropagated to the generator for target images (we want the encoder to learn to map the target images to the same distribution as the source images, not vice versa). Likewise, the translation classification loss ( $Q_{trc}$ ) is only backpropagated to the second encoder and classifier ( $f_y$  and  $h$ ). This prevents the translator ( $g_y$ ) from cheating by hiding class information in the translated images.

### 4.1. MNIST, USPS, and SVHN digits datasets

First, we demonstrate our method on domain adaptation between three digit classification datasets, namely MNIST [18], USPS [14], and the Street View House Numbers (SVHN) [25] datasets. We followed the experimental protocol of [5, 34, 20, 35, 30] where we treated one of the digit datasets as a labeled source domain and another dataset as unlabeled target domain. We trained our framework for adaptation from MNIST  $\rightarrow$  USPS, USPS  $\rightarrow$  MNIST, and SVHN  $\rightarrow$  MNIST. Figure 3 shows examples of MNIST to SVHN input and translated images.

For a fair comparison with previous methods, our feature extractor network (encoder,  $f_x$  and  $f_y$ ) is a modified version of LeNet [18]. Our decoders (i.e.  $g_x$  and  $g_y$ ) con-

Method							MNIST $\rightarrow$ USPS	USPS $\rightarrow$ MNIST	SVHN $\rightarrow$ MNIST
Source only							24.8	42.9	39.7
Gradient reversal [5]							28.9	27.0	26.1
Domain confusion [34]							20.9	33.5	31.9
CoGAN [20]							8.8	10.9	-
ADDA [35]							10.6	9.9	24.0
DTN [33]							-	-	15.6
WDAN [38]							27.4	34.6	32.6
PixelDA [1]							4.1	-	-
DRCN [6]							8.2	26.3	18.0
Gen to Adapt [30]							7.5	9.2	15.3
$\lambda_z$	$\lambda_{tr}$	$\lambda_{id_X}$	$\lambda_{id_Y}$	$\lambda_{cyc}$	$\lambda_{trc}$	Ours			
							8.9	33.0	28.5
✓							2.1	2.8	19.9
			✓				2.1	28.5	23.7
			✓		✓		1.4	22.6	29.5
✓			✓				1.3	3.0	12.1
✓			✓		✓		<b>1.2</b>	<b>2.4</b>	<b>9.9</b>
✓	✓	✓	✓		✓		2.5	3.6	10.0
✓	✓	✓	✓	✓	✓		1.5	2.6	10.4

Table 2. Performance of various prior methods as well as ours and ablations on digits datasets domain adaptation. MNIST  $\rightarrow$  USPS indicates MNIST is the source domain (labels available) and USPS is the target domain (no labels available). Results reported are classification error rate (lower is better). Blue is best prior method, bold is best overall. Our results are considerably better than the prior state of the art.

Method						A $\rightarrow$ W	W $\rightarrow$ A	A $\rightarrow$ D	D $\rightarrow$ A	W $\rightarrow$ D	D $\rightarrow$ W
Domain confusion [36]						61.8	52.2	64.4	21.1	98.5	95.0
Transferable Features [21]						68.5	53.1	67.0	54.0	99.0	96.0
Gradient reversal [5]						72.6	52.7	67.1	54.5	99.2	96.4
DHN [37]						68.3	53.0	66.5	55.5	98.8	96.1
WDAN [38]						66.8	52.7	64.5	53.8	98.7	95.9
DRCN [6]						68.7	<b>54.9</b>	66.8	<b>56.0</b>	99.0	96.4
$\lambda_z$	$\lambda_{tr}$	$\lambda_{id_X}$	$\lambda_{id_Y}$	$\lambda_{trc}$	Ours						
						59.1	46.4	61.0	45.3	98.0	92.8
✓						70.8	49.0	67.1	43.4	98.2	90.8
			✓			61.1	49.6	67.3	49.8	99.0	94.7
✓			✓	✓		71.2	49.1	70.9	45.5	97.8	94.3
✓	✓	✓	✓	✓		<b>75.3</b>	52.1	<b>71.1</b>	50.1	<b>99.6</b>	<b>96.5</b>

Table 3. Accuracy (larger is better) of various methods on the Office dataset consisting of three domains: Amazon (A), Webcam (W) and DSLR (D). A  $\rightarrow$  W indicates Amazon is the source domain (labels available) and Webcam is the target domain (no labels available). Bold is best. Our method performs best on 4 out of 6 of the tasks.

sist of three transposed convolutional layers. Our image discriminators consist of three convolutional layers and our feature discriminator consists of three fully connected layers. All images from MNIST and USPS were bilinearly upsampled to 32x32. Images from SVHN were converted to gray scale. We also included very simple data augmentation in the form of random translations and rotations as it helped a lot. For our hyperparameters we used:  $\lambda_c = 1.0$ ,  $\lambda_z = 0.05$ ,  $\lambda_{id} = 0.1$ ,  $\lambda_{tr} = 0.02$ ,  $\lambda_{cyc} = 0.1$ ,  $\lambda_{trc} = 0.1$ .

We compare our method to nine prior works (see Table. 2). Our method consistently out performs the prior state of the art by a significant margin. We also show ab-

lations to analyze how much each of the loss terms contributes to the overall performance. First note that even our results without domain adaption (top row of Table. 2 below the line) are better than many prior methods. This is purely due to the simple data augmentation. Next we see that  $\lambda_z$  and  $\lambda_{id_Y}$  both improve results a lot. However the combination of them with  $\lambda_{trc}$  produces the best results. This is especially apparent in SVHN to MNIST, which is the most challenging. Finally, in this case, the remaining losses don't help any further.

Figure 3 A,B,C show TSNE embeddings of the features extracted from the source and target domain when trained

Method	road	sidewalk	building	wall	fence	pole	t light	t sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Source only	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	21.1
FCNs in the Wild [12]	67.4	29.2	64.9	15.6	8.4	12.4	9.8	2.7	74.1	12.8	66.8	38.1	2.3	63.0	9.4	5.1	0.0	3.5	0.0	27.1
Ours	<b>85.3</b>	<b>38.0</b>	<b>71.3</b>	<b>18.6</b>	<b>16.0</b>	<b>18.7</b>	<b>12.0</b>	<b>4.5</b>	72.0	<b>43.4</b>	63.7	<b>43.1</b>	<b>3.3</b>	<b>76.7</b>	<b>14.4</b>	<b>12.8</b>	<b>0.3</b>	<b>9.8</b>	<b>0.6</b>	<b>31.8</b>
Source only - DenseNet	67.3	23.1	69.4	13.9	14.4	21.6	<b>19.2</b>	<b>12.4</b>	78.7	24.5	74.8	49.3	3.7	54.1	8.7	5.3	2.6	6.2	<b>1.9</b>	29.0
Ours - DenseNet	<b>85.8</b>	<b>37.5</b>	<b>80.2</b>	<b>23.3</b>	<b>16.1</b>	<b>23.0</b>	14.5	9.8	<b>79.2</b>	<b>36.5</b>	<b>76.4</b>	<b>53.4</b>	<b>7.4</b>	<b>82.8</b>	<b>19.1</b>	<b>15.7</b>	<b>2.8</b>	<b>13.4</b>	1.7	<b>35.7</b>

Table 4. Performance (Intersection over Union) of various methods on driving datasets domain adaptation. Above the line uses the standard dilated ResNet as the encoder. Our method performs the best overall and on all sub categories except two. Switching to a DenseNet encoder beats the previous method even without domain adaptation. DenseNet plus our method significantly out performs the previous method. Blue is best with ResNet, Bold is best overall.

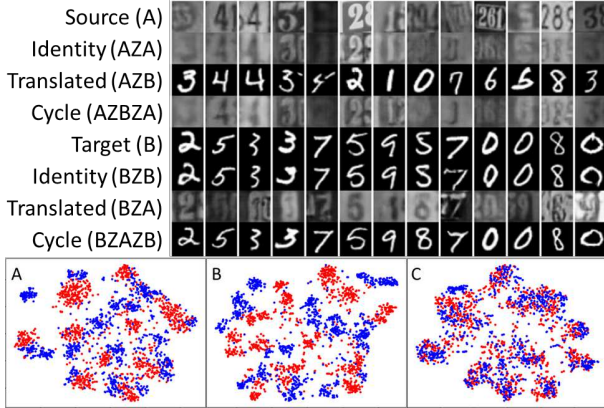


Figure 3. Top) Image to image translation examples for MNIST to SVHN. Bottom) TSNE embedding visualization of the latent space. Red are source images, Blue are target images. A) No adaptation. B) Image to image adaptation without latent space discriminator. C) Full adaptation.

without adaptation, with image to image loss only, and our full model. It can be seen that without adaptation, the source and target images get clustered in the feature space but the distributions do not overlap which is why classification fails on the target domain. Just image to image translation is not enough to force the distributions to overlap as the networks learn to map source and target distributions to different areas of the feature space. Our full model includes a feature distribution adversarial loss, forcing the source and target distributions to overlap, while image translation makes the features richer yielding the best adaptation results.

#### 4.2. Office dataset

The Office dataset [29] consists of images from 31 classes of objects in three domains: Amazon (A), Webcam (W) and DSLR (D) with 2817, 795 and 498 images respectively. Our method performs the best in four out of six of the tasks (see Table 3). The two tasks that ours is not best at consist of bridging a large domain shift with very little training data in the source domain (795 and 498 respectively). Here the ablations show that the translation loss

( $Q_{tr}$ ) helps.

For our encoder we use a ResNet34 pretrained on ImageNet. The encoder is trained with a smaller learning rate ( $2 \times 10^{-5}$ ), to keep the weights closer to their good initialization. Images are down sampled to 256x256 and then a random crop of size 224x224 is extracted. The final classification layer is applied after global average pooling. Our decoders consist of 5 stride 2 transposed convolutional layers. The image discriminators consist of 4 stride 2 convolutional layers. The feature discriminator consists of 3 1x1 convolutions followed by global average pooling.

Our hyperparameters were:  $\lambda_c = 1.0$ ,  $\lambda_z = 0.1$ ,  $\lambda_{tr} = 0.005$ ,  $\lambda_{id} = 0.2$ ,  $\lambda_{cyc} = 0.0$ ,  $\lambda_{trc} = 0.1$ .

#### 4.3. GTA5 to Cityscapes

We also demonstrate our method for domain adaptation between the synthetic (photorealistic) driving dataset GTA5 [26] and the real dataset Cityscapes [2]. The GTA5 dataset consists of 24,966 densely labeled RGB images of size  $1914 \times 1052$ , containing 19 classes that are compatible with the Cityscapes dataset (See Table 4). The Cityscapes dataset contains 5,000 densely labeled RGB images of size  $2040 \times 1016$  from 27 different cities. Here the task is pixel level semantic segmentation. Following the experiment in [12], we use the GTA5 images as the labeled source dataset and the Cityscapes images as the unlabeled target domain.

We point out that the convolutional networks in our model are interchangeable. We include results using a dilated ResNet34 encoder for fair comparison with previous work, but we found from our experiments that the best performance was achieved by using our new Dilated Densely-Connected Networks (i.e. Dilated DenseNets) for the encoders which are derived by replacing strided convolutions with dilated convolutions [39] in the DenseNet architecture [13]. DenseNets have previously been used for image segmentation [16] but their encoder/decoder structure is more cumbersome than what we proposed. We use a series of transposed convolutional layers for the decoders.

Our decoders consist of a stride 1 convolutional layer followed by 3 stride 2 transposed convolutional layers. The



image discriminators consist of 4 stride 2 convolutional layers. We did not include the cycle consistency constraint due to memory issues. Due to computational and memory constraints, we down sample all images by a factor of two prior to feeding them into the networks. Output segmentations are bilinearly up sampled to the original resolution. We train our network on 256x256 patches of the down sampled images, but test on the full (downsampled) images convolutionally. Our hyperparameters were:  $\lambda_c = 1.0$ ,  $\lambda_z = 0.01$ ,  $\lambda_{tr} = 0.04$ ,  $\lambda_{id} = 0.2$ ,  $\lambda_{cyc} = 0.0$ ,  $\lambda_{trc} = 0.1$ .

Our encoder architecture (dilated ResNet/DenseNet) is optimized for segmentation and thus it is not surprising that our translations (see Figure. 4) are not quite as good as those reported in [40]. Qualitatively, it can be seen from Figure 4 that our segmentations are much cleaner compared to no adaptation. Quantitatively (see Table 4), our method outperforms the previous method [12] on all categories except 3, and is 5% better overall. Furthermore, we show that using Dilated DenseNets in our framework, increases the SOA by 8.6%.

## 5. Conclusion

We have proposed a general framework for unsupervised domain adaptation which encompasses many recent works as special cases. Our proposed method simultaneously achieves image to image translation, source discrimination, and domain adaptation.

Our implementation outperforms state of the art on adaptation for digit classification and semantic segmentation of driving scenes. When combined with the DenseNet architecture our method significantly outperforms the current state of the art.

## Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X Pascal GPU used for this research.

## References

- [1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3722–3731, 2017. 6
- [2] 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. 2, 7
- [3] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analy-

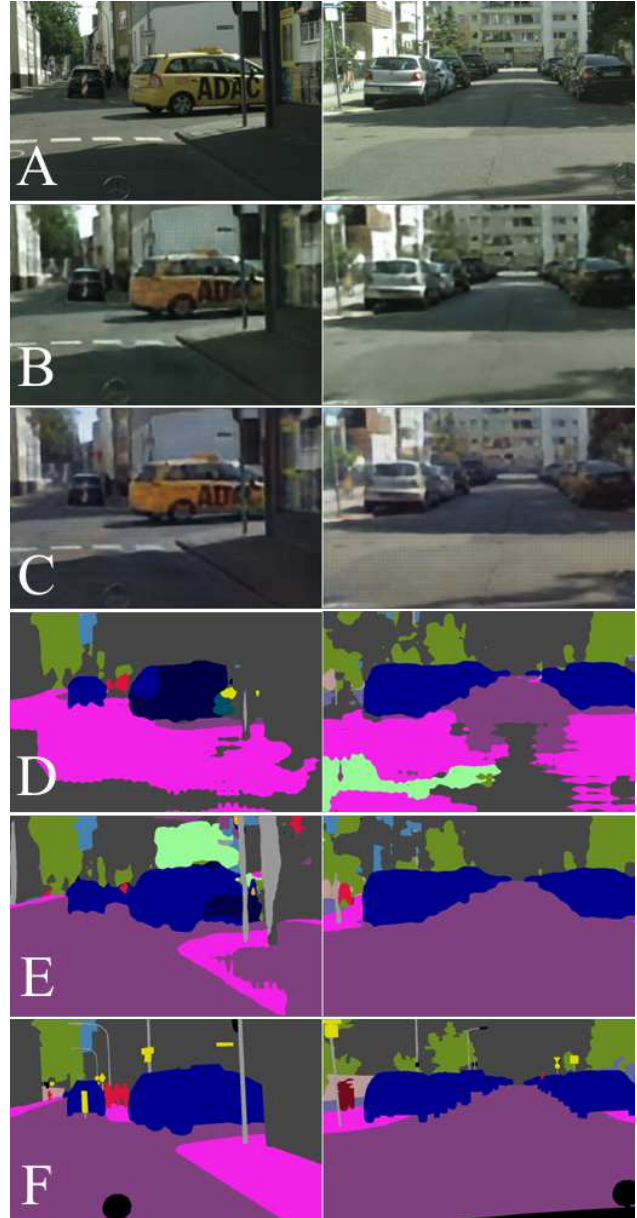


Figure 4. A) Input image from real Cityscapes dataset. B) Identity mapped image. C) Translated image. D) Segmentation without domain adaptation. E) Our Segmentation. F) Ground truth. Although our image translations might not be as visually pleasing as those in [40] (our architecture is not optimized for translation), they succeed in their goal of domain adaptation.

sis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016. 2

- [4] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 2



- [5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. 2, 5, 6
- [6] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2, 3, 5, 6
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5
- [8] A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. 2009. 1, 2
- [9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 5
- [10] 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*, pages 770–778, 2016. 1
- [11] 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. 4
- [12] 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. 2, 3, 5, 7, 8
- [13] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016. 1, 7
- [14] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994. 5
- [15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 3
- [16] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017. 7
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5
- [19] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 4
- [20] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 5, 6
- [21] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 2, 6
- [22] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016. 2
- [23] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei. Label efficient learning of transferable representations across domains and tasks. In *Conference on Neural Information Processing Systems (NIPS)*, 2017. 2
- [24] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016. 5
- [25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011. 5
- [26] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016. 2, 7
- [27] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 2
- [28] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Moressi, F. Cole, and K. Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. *arXiv preprint arXiv:1711.05139*, 2017. 3
- [29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010. 7
- [30] 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, 5, 6
- [31] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016. 2
  - [32] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. 2
  - [33] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 6
  - [34] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. 2, 5, 6
  - [35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017. 2, 3, 5, 6
  - [36] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2, 3, 6
  - [37] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5385–5394. IEEE, 2017. 6
  - [38] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2281, 2017. 6
  - [39] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. *arXiv preprint arXiv:1705.09914*, 2017. 7
  - [40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 2, 3, 5, 8