DRANet: Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation

Seunghun Lee
DGIST
lsh5688@dgist.ac.kr

Sunghyun Cho
POSTECH CSE & GSAI
s.cho@postech.ac.kr

Sunghoon Im*
DGIST
sunghoonim@dgist.ac.kr

Abstract

In this paper, we present DRANet, a network architecture that disentangles image representations and transfers the visual attributes in a latent space for unsupervised cross-domain adaptation. Unlike the existing domain adaptation methods that learn associated features sharing a domain, DRANet preserves the distinctiveness of each domain's characteristics. Our model encodes individual representations of content (scene structure) and style (artistic appearance) from both source and target images. Then, it adapts the domain by incorporating the transferred style factor into the content factor along with learnable weights specified for each domain. This learning framework allows bi-/multi-directional domain adaptation with a single encoder-decoder network and aligns their domain shift. Additionally, we propose a content-adaptive domain transfer module that helps retain scene structure while transferring style. Extensive experiments show our model successfully separates content-style factors and synthesizes visually pleasing domain-transferred images. The proposed method demonstrates state-of-the-art performance on standard digit classification tasks as well as semantic segmentation tasks.

1. Introduction

The use of deep neural networks (DNN) has led to significant performance improvements in a variety of areas, including computer vision [6], machine learning [13], and natural language processing [7]. However, problems remain, particularly domain gaps between data, which can significantly degrade model performance. Extensive efforts have been made to generalize the models across domains using unsupervised domain adaptation [1, 38, 23, 36, 9, 32, 37, 21, 2, 15, 39]. Unsupervised domain adaptation attempts to align the distribution shift in labeled source data with unlabeled target data. Various strategies have been explored to bridge the gap across domains, for example, by feature learning and generative pixel-level adaptation.

---

*Corresponding author.

(a) Traditional domain adaptation [9, 15]

(b) Linear feature separation [42] and domain adaptation

(c) Our feature separation and domain adaptation (*DRANet*)
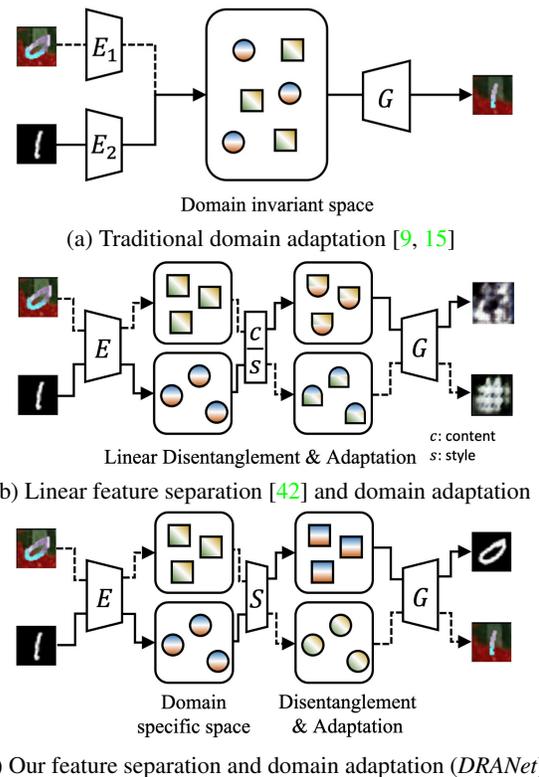
Figure 1. Illustration of *DRANet* and the competitive methods (domain adaptation [9, 15], representation disentanglement [42]). Note that E, S, and G are an encoder, a separator, and a generator.

Feature-level methods [38, 23, 32, 36, 9, 32, 37] learn features that combine task-discrimination and domain-invariance, where both domains are mapped into a common feature space. Domain invariance typically involves minimizing some feature distance metric [38, 23, 32] or adversarial discriminator accuracy [9]. Pixel-level approaches [21, 2] perform a similar distribution alignment, not in a feature space but in the raw pixel space by leveraging the power of Generative Adversarial Networks (GANs) [14, 24, 28, 30, 4]. They adapt source domain images so that they appear as if drawn from the target domain.

Some studies [15, 35, 39] incorporate both pixel-level and feature-level approaches to achieve complementary benefits.

Recently, the field of study has been further advanced by learning disentangled representations into the exclusive and shared components in a latent feature space [3, 12, 22, 45]. They demonstrate that representation disentanglement improves a model's ability to extract domain invariant features, as well as the domain adaptation performance. However, these methods still focus on the associated features between two domains such as shared and exclusive components, so they require multiple encoders and generators specialized in individual domains. Moreover, the network training relies heavily on a task classifier with ground-truth class labels, in addition to domain classifiers.

To tackle these issues, we propose DRANet, a single feed-forward network, that does not require any ground-truth task labels for cross-domain adaptation. In contrast to previous approaches in Fig. 1-(a) that map all domain images into a shared feature space, we focus on extracting the domain-specific features that preserve individual domain characteristics in Fig. 1-(c). Then, we disentangle the discriminative features of individual domains into the content and style components using a separator, which are later used to generate the domain-adaptive features. Unlike the previous feature separation work [42], which linearly divides latent vectors into two components in Fig. 1-(b), our separator is tailored to disentangle latent variables in a nonlinear manifold. Our intuition behind the network design is that different domains may have different distributions for their contents and styles, which cannot be effectively handled by the linear separation of latent vectors. Thus, to handle such difference, our network adopts the non-linear separation and domain-specific scale parameters that are dedicated to handle such inter-domain difference.

To the best of our knowledge, DRANet is the first approach based solely on the individual domain characteristics for unsupervised cross-domain adaptation. It enables us to apply a single encoder-decoder network for a multi-directional domain transfer from fully unlabeled data. The distinctive points of our approach are summarized as follows:

- We present DRANet, which disentangles image representation and adapts the visual attributes in a latent space to align the domain shift.
- We propose a content-adaptive domain transfer module that helps to synthesize realistic images of complex segmentation datasets, such as CityScapes [5] and GTA5 [29].
- We demonstrate that images synthesized by our approach boost the task performances and achieve state-of-the-art performance on standard digit classification tasks as well as semantic segmentation tasks.

2. Related Work

2.1. Unsupervised Domain Adaptation

Feature-level domain adaptation methods typically align learning distribution by modifying the discriminative representation space. The strategy is to guide feature learning by minimizing the difference between the feature space statistics of the source and target. Early deep adaptive approaches minimize some measurements of domain shift such as maximum mean discrepancy [38, 23] or correlation distances [32]. Recent works [9, 36, 37] learn the representation that is discriminative of source labels while not being able to distinguish between domain using an adversarial loss inspired by the work [1]. The domain-invariant features are discovered using standard backpropagation training with minimax loss [9], domain confusion loss [36], or GAN loss [37].

Another approach to unsupervised domain adaptation is the generative pixel-level domain adaptation, which synthesizes images with the content of source images and the style of target images using the adversarial training [14]. Liu and Tuzel [21] accomplish to learn the joint distribution of source and target representations by weight sharing, using a specific layer responsible for decoding abstract semantics. Bousmalis *et al*. [2] use GANs to learn transformations in the pixel space from one domain to another. Hoffman *et al*. [15] adapt representations both at the pixel and feature levels while enforcing both structural and semantic consistency using a cycle-consistency loss. Ye *et al*. [39] also incorporate both pixel and feature-level domain classifiers to calibrate target domain images whose representations are close to those of the source domain.

2.2. Disentangling Internal Representation

The separation of style and content components in a latent space has been widely studied for artistic style transfer [33, 8, 11, 42, 43]. Tenebaum and Freeman [33] show how perceptual systems can separate the content and style factors, and propose bilinear models to solve these two factors. Elgammal and Lee [8] introduce a method to separate style and content on manifolds representing dynamic objects. Gatys *et al*. [11] show how generic feature representations learned by a CNN manipulate the content and style of natural images. Zhang *et al*. [43] propose a neural network representing each style and content with a small set of images, while separating the representations. Zhang *et al*. [42] bimodally divide feature representations into the content and style components.

Among the studies on domain adaptation, the search for approaches to disentangling internal representations has recently grown in interest. Bousmalis *et al*. [3] learn to extract image representations that are partitioned into two subspaces: private and shared components and show that
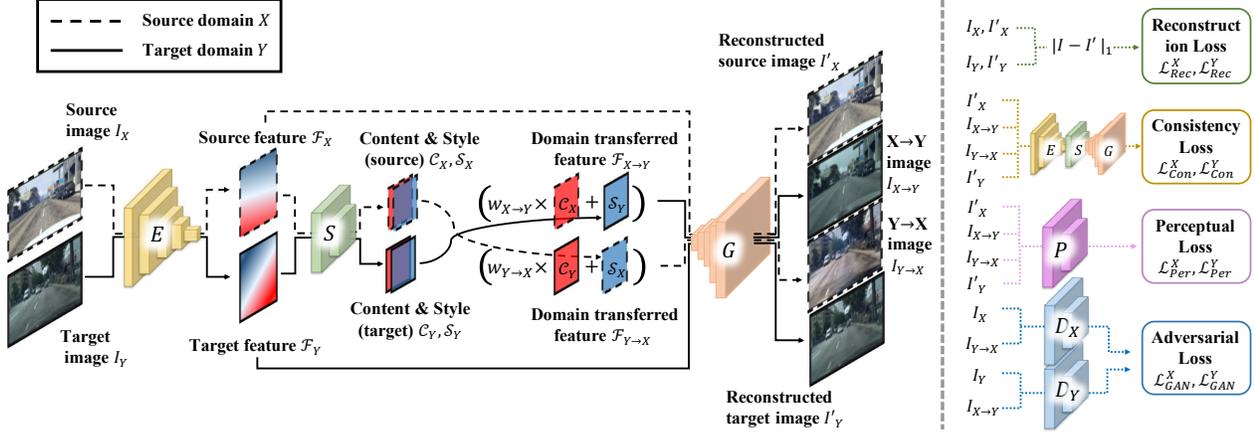
Figure 2. Overview of our model. (Left) Image translation blocks involving an encoder E, a separator S, and a generator G. The source and target images I_X, I_Y are the input, and the reconstructed images I'_X, I'_Y and domain transferred images $I_{X \to Y}, I_{Y \to X}$ are the output. (Right) The training losses involving reconstruction $\mathcal{L}_{Rec}$, consistency $\mathcal{L}_{Con}$, perceptual $\mathcal{L}_{Per}$, and adversarial $\mathcal{L}_{GAN}$ loss.

the modeling of unique features helps to extract domain-invariant features. Gonzalez-Garcia *et al.* [12] attempt to disentangle factors that are exclusive in both domains, and factors that are shared across domains. Liu *et al.* [22] propose a cross-domain representation disentangler that bridges the information across data domains and transfers the attributes. Zou *et al.* [45] introduce a joint learning framework that separates id-related/unrelated features for person re-identification tasks. We discuss the major differences between our work and the listed works in Sec. 1.

3. DRANet

3.1. Overview

The overall pipeline of our method is illustrated in Fig. 2. Our framework can be extended to domain transfer across three domains, as shown in Fig. 3, although the example only shows two domain case for simple illustration. The networks consist of an encoder E, a feature separator S, a generator G, two discriminators of the source and target domains D_X, D_Y, and a perceptual network P. In the training phase, we learn all of the parameters of these networks, as well as the feature scaling factors $w_{X \to Y}, w_{Y \to X}$ which compensate for the distribution of two domains. Given the source and target images I_X, I_Y, the encoder E extracts the individual features $\mathcal{F}_X, \mathcal{F}_Y$ that later pass through the generator G to reconstruct the original input images I'_X, I'_Y. The separator S disentangles each feature $\mathcal{F}_X, \mathcal{F}_Y$ into the components of scene structure and artistic appearance, which in this paper we call the content $\mathcal{C}_X, \mathcal{C}_Y$ and the style $\mathcal{S}_X, \mathcal{S}_Y$, respectively. Then, the transferred domain features $\mathcal{F}_{X \to Y}, \mathcal{F}_{Y \to X}$ are synthesized with the learnable scale parameters $w_{X \to Y}, w_{Y \to X}$. The generator G maps the original features $\mathcal{F}_X, \mathcal{F}_Y$ and the transferred features $\mathcal{F}_{X \to Y}, \mathcal{F}_{Y \to X}$ into their image space

$I'_X, I'_Y, I_{X \to Y}, I_{Y \to X}$, respectively. The pretrained perceptual network P, extracts perceptual features to impose the constraints on both content similarity and style similarity. We use two discriminators, D_X, D_Y, to impose the adversarial loss on both domains. In the test phase, just the encoder E, the separator S, the generator G, and domain weights w are used to produce domain transferred images $I_{X \to Y}, I_{Y \to X}$ given source and target images I_X, I_Y. With the single feed-forward network E-S-G, our method enables the bi-directional domain transfer of input images.

3.2. Disentangling Representation and Adaptation

In this subsection, we describe the motivation for the design of our separator S. We first extract the individual image features $\mathcal{F}_X, \mathcal{F}_Y$ using the weight-shared encoder:

$$\mathcal{F}_X = E(I_X), \ \mathcal{F}_Y = E(I_Y). \tag{1}$$

The separator disentangles these features into scene structure and artistic appearance factors. We hypothesize that the nonlinear manifold learning is still necessary to map each domain-specific representation into the content or style spaces as demonstrated in [8]. Thus, we learn a non-linear projection function S that separates the features $\mathcal{F}_X$ into content $\mathcal{C}_X$ and style $\mathcal{S}_X$ factors, as follows:

$$\mathcal{C}_X = w_X S(\mathcal{F}_X), \ \mathcal{S}_X = \mathcal{F}_X - w_X S(\mathcal{F}_X), \tag{2}$$

where w_X is the weight parameter that normalizes the distribution of content space, which helps to compensate for the distribution shift. The content component is obtained using the non-linear function and the learnable feature scaling parameters, while the style component is defined by subtracting content components from the whole feature. The target representation $\mathcal{F}_Y$ is also passed through the same separator S, and outputs the target content and style $\mathcal{C}_Y, \mathcal{S}_Y$,

Figure 3. Tri-directional domain adaptation results from our single network. DRANet keeps the content of source images and transfers the domain of target images.

but for simplicity here we only denote the source domain case.

The disentangled representation is used to transfer the domain of features across domains as follows:

$$\mathcal{F}_{X\to Y} = w_{X\to Y}\mathcal{C}_X + \mathcal{S}_Y, \ \ \mathcal{F}_{Y\to X} = w_{Y\to X}\mathcal{C}_Y + \mathcal{S}_X,$$
$$\text{where } w_{X\to Y} = \frac{w_Y}{w_X}, \ w_{Y\to X} = \frac{w_X}{w_Y}. \tag{3}$$

In our implementation, we directly learn the relative scale parameters $w_{X\to Y}, w_{Y\to X}$ along with all model parameters. Finally, we pass all representations involving the domain adaptive features $\mathcal{F}_{X\to Y}, \mathcal{F}_{Y\to X}$ and the original source and target features $\mathcal{F}_X, \mathcal{F}_Y$ through the generator G to project them into image space as follows:

$$I_{X\to Y} = G(\mathcal{F}_{X\to Y}), \ I_{Y\to X} = G(\mathcal{F}_{Y\to X}),$$
$$I'_X = G(\mathcal{F}_X), \ I'_Y = G(\mathcal{F}_Y), \tag{4}$$

where $I_{X\to Y}, I_{Y\to X}$ are the domain adapted images and I'_X, I'_Y are the reconstructed images.

3.3. Content-Adaptive Domain Transfer (CADT)

Style transfer tends to struggle with the complex scenes containing various objects, such as a driving scene. This is because those images are composed of different scene structure, as well as various object composition. To tackle this problem, we present a Content-Adaptive Domain Transfer (CADT). The key idea of this module is to search the target features whose content component is most similar to the source features. Then, the domain transfer is conducted by reflecting more style information from more suitable target features. To achieve this, we design a content similarity ma-

trix for the database in a mini-batch, as follows:

$$\mathcal{H}_{row} = \sigma_{row}\left(\mathcal{C}_X \cdot \mathcal{C}_Y^\top\right) = \begin{bmatrix} \mathcal{C}_{11} & \cdots & \mathcal{C}_{1b} \\ \vdots & \ddots & \vdots \\ \mathcal{C}_{b1} & \cdots & \mathcal{C}_{bb} \end{bmatrix}, \tag{5}$$
$$\mathcal{C}_X, \mathcal{C}_Y \in \mathbb{R}^{B\times N},$$

where σ_{row} is the softmax operation in the row dimension. The size of the content factors $\mathcal{C}_X$ is defined by the batch size B and the feature dimension N. The matrix $\mathcal{H}_{row}$ contains information about the level of similarity between components in the mini-batch. Based on the similarity matrix, we build a content-adaptive style feature as follows:

$$\hat{\mathcal{S}}_Y = \mathcal{H}_{row}\mathcal{S}_Y, \text{where } \mathcal{S}_Y \in \mathbb{R}^{B\times N}. \tag{6}$$

More visually pleasing results can be expected than when using the normal transferring method because the content features are more likely to be stylized by the scenes containing similar structure and object composition. We empirically demonstrate this in Fig. 8. To apply the content-adaptive domain transfer in the opposite direction, the content similarity matrix is simply obtained:

$$\mathcal{H}_{col} = \left(\sigma_{col}\left(\mathcal{C}_X \cdot \mathcal{C}_Y^\top\right)\right)^\top, \tag{7}$$

where σ_{col} is the softmax in the column direction.

3.4. Training Loss

We train our network with an encoder E, a separator S, and a generator G by minimizing the loss function $\mathcal{L}^d$ while the discriminator D_d tries to maximize it:

$$\min_{E,S,G}\left(\sum_{d\in\{X,Y\}} \max_{D_d} \mathcal{L}^d\right), \tag{8}$$

where the domain d is either a source or target domain X, Y. The overall loss of our framework consists of the reconstruction $\mathcal{L}_{Rec}$, consistency $\mathcal{L}_{Con}$, perceptual $\mathcal{L}_{Per}$, and adversarial $\mathcal{L}_{GAN}$ loss with the balancing term α_i:

$$\mathcal{L}^d = \alpha_1\mathcal{L}_{Rec}^d + \alpha_2\mathcal{L}_{GAN}^d + \alpha_3\mathcal{L}_{Con}^d + \alpha_4\mathcal{L}_{Per}^d. \tag{9}$$

The followings are the details of each loss.

Reconstruction Loss. We impose an L1 loss to learn E and G that minimizes the difference between input image I_d and the reconstructed image I'_d:

$$\mathcal{L}_{Rec}^d = \mathcal{L}_1(I_d, I'_d), \text{ where } I'_d = G(E(I_d)). \tag{10}$$

Adversarial Loss. We apply two discriminators $D_{d\in\{X,Y\}}$ to evaluate the adversarial loss on the source and target domain, respectively. The following is the adversarial loss for the domain adaptation of X to Y:

$$\mathcal{L}_{GAN}^Y = \mathbb{E}_{y\sim p_{data(Y)}}\left[\log D_Y(y)\right]$$
$$+ \mathbb{E}_{(x,y)\sim p_{data(X,Y)}}\left[\log(1 - D_Y(I_{X\to Y}(x,y))\right]. \tag{11}$$

(a) Source and target images   (b) Domain transferred images

Figure 4. Various domain transferred examples from MNIST to MNIST-M. (a) Top-left image is source image of digit 2 and the others are target images. (b) Domain transferred images.

We impose the same adversarial loss $\mathcal{L}_{GAN}^X$ for the adaptation of Y to X as well. We apply spectral normalization [25] to all layers in G and D, and use PatchGAN Discriminator [17] with the hinge version of adversarial loss [20, 34, 26, 41] for driving scene adaptation.

Consistency Loss. The consistency loss attempts to retain the content and style components after re-projecting the domain transferred images into the representation space denoted as:

$$\mathcal{L}_{Con}^X = \mathcal{L}_1\big(\mathcal{C}_X, \mathcal{C}_{X \to Y}\big) + \mathcal{L}_1\big(\mathcal{S}_X, \mathcal{S}_{Y \to X}\big),$$
$$\mathcal{L}_{Con}^Y = \mathcal{L}_1\big(\mathcal{C}_Y, \mathcal{C}_{Y \to X}\big) + \mathcal{L}_1\big(\mathcal{S}_Y, \mathcal{S}_{X \to Y}\big), \quad (12)$$

where the content $C_{X \to Y}, C_{Y \to X}$ and style $S_{X \to Y}, S_{Y \to X}$ factors are extracted by passing the domain transferred images $I_{X \to Y}, I_{Y \to X}$, respectively, through the same encoder E and separator S. This loss explicitly encourages the scene structure consistency and artistic appearance consistency before and after domain adaptation.

Perceptual Loss. Conventionally, the GT class labels in (semi-)supervised training are provided as the semantic cues guiding the representation disentanglement. However, our framework trains disentangling representations without any labeled data. To learn the disentangler in an unsupervised manner, we impose a perceptual loss [18] which is widely known as a typical framework for style transfer, defined as:

$$\mathcal{L}_{Per}^X = \mathcal{L}_{Content}^X + \lambda \mathcal{L}_{Style}^X,$$
$$\mathcal{L}_{Per}^Y = \mathcal{L}_{Content}^Y + \lambda \mathcal{L}_{Style}^Y, \quad (13)$$

where $\mathcal{L}_{Content}^X, \mathcal{L}_{Content}^Y$ are the content losses, and $\mathcal{L}_{Style}^X, \mathcal{L}_{Style}^Y$ are the style losses defined as:

$$\mathcal{L}_{Content}^Y = \sum_{l \in L_C} \| P_l(I_X) - P_l(I_{X \to Y}) \|_2^2,$$
$$\mathcal{L}_{Style}^Y = \sum_{l \in L_S} \| \mathcal{G}(P_l(I_Y)) - \mathcal{G}(P_l(I_{X \to Y})) \|_F^2, \quad (14)$$

where the set of layers L_C, L_S are the subset of the perceptual network P. The weight parameter λ balances the
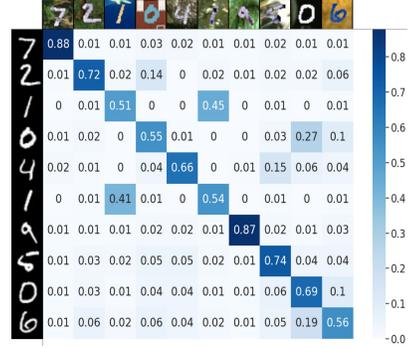
Figure 5. Content similarity between MNIST and MNIST-M.

two losses, and $\mathcal{G}$ is the function that builds a Gram Matrix, given the features of each layer l [10]. We also apply batch-instance normalization [27] for better stylization. Details of architecture are described in the supplementary materials.

4. Experiments

We evaluate DRANet for unsupervised domain adaptation on digit classification in Sec. 4.1 and driving scene segmentation in Sec. 4.2. We compare our bi-/tri-directional domain transfer results against multiple state-of-the-art un-/semi-supervised domain adaptation methods. We also conduct an extensive ablation study to demonstrate the effectiveness of each proposed module in Sec. 4.3. For the evaluations, we use the standard split of training and test sets the same as the existing unsupervised domain adaptations [2, 39]. We train a task-classifier using stylized source training sets produced by DRANet and evaluate its performance on the target domain test sets. We describe the training details in the supplementary materials.

4.1. Adaptation for Digit Classification

Unlike existing domain adaptation methods, where a single model is responsible for domain transfer in one direction, our single model is able to deal with multi-directional domain adaption. We demonstrate the versatility of DRANet by transferring images across the multiple domains using three digit datasets: MNIST [19], MNIST-M [9], and USPS [16]. We train our model for bi-directional domain adaptation (MNIST to MNIST-M or USPS, and its opposite direction) as shown in Fig. 2. We also train the adaptation model tri-directionally (MNIST to MNIST-M and USPS, and their opposite directions) and show the results in Fig. 3. Note that we have not explicitly transferred the domain between MNIST-M and USPS during training, but the results show that DRANet is also applicable for the adaptation between them.

As shown in Tab. 1, our model, either trained for two or three domains, outperforms all the competitive methods [39, 15, 2, 21, 37, 3, 9]. The results also show that our model even achieves higher performance than the model

Method	MNIST to USPS	USPS to MNIST	MNIST to MNIST-M	MNIST-M to MNIST
Source Only	80.2	44.9	62.5	97.8
DANN [9]	85.1	73.0	77.4	-
DSN [3]	91.3	-	83.2	-
ADDA [37]	90.1	95.2	-	-
CoGAN [21]	91.2	89.1	62.0	-
pixelDA [2]	95.9	-	98.2	-
CyCADA [15]	95.6	96.5	-	-
LC + CycleGAN [39, 44]	97.1	**98.3**	-	-
Ours (Bi-directional)	**98.2**	97.8	**98.7**	**99.3**
Ours (Tri-directional)	97.6	96.9	98.3	99.0
Target Only	97.8	99.1	96.2	99.1

Table 1. Result comparison of DRANet to state-of-the-art methods on domain adaptation for digit classification. We report the performance from both bi-directional and tri-directional domain adaptation. Note that ours(bi-directional) and ours(tri-directional) use two models (MNIST-USPS, MNIST-MNISTM) and a model (MNIST-USPS-MNISTM), respectively to evaluate all four domain adaptation tasks.

(a) GTA5 original images

(b) Transferred images using (a) GTA5 content and (c) CityScapes style.

(c) CityScapes original images

(d) Transferred images using (c) CityScapes content and (a) GTA5 style.
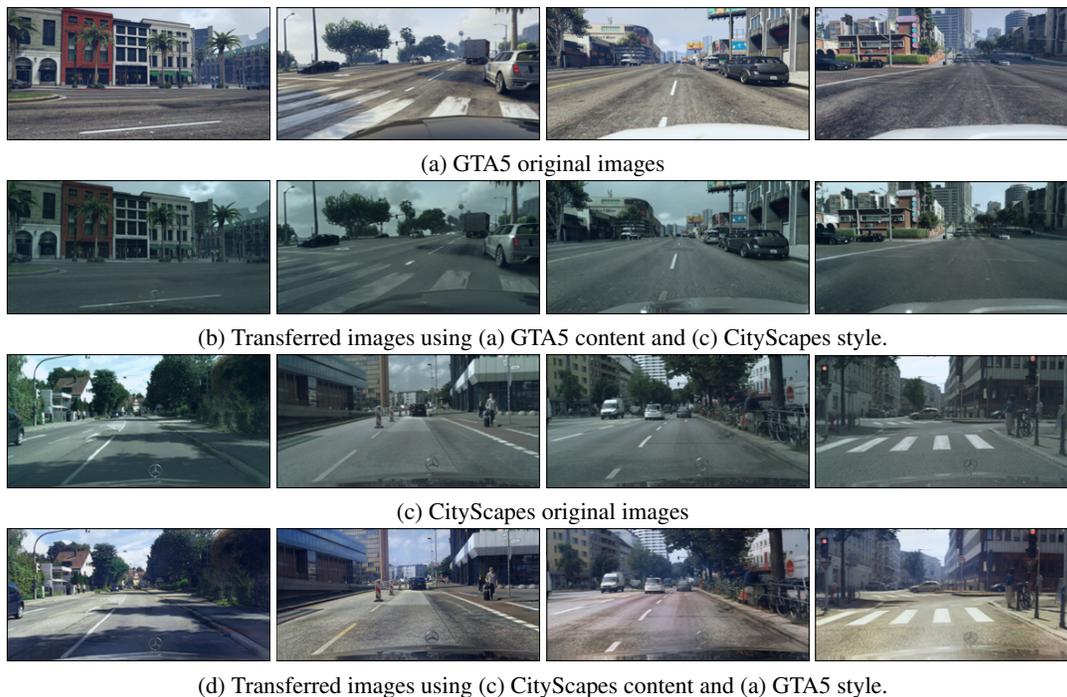
Figure 6. Domain adaptation results from our single DRANet in driving scenes. (a), (c) Original images. (b), (d) Transferred images.

trained only on target except for the experiment of USPS to MNIST. This is because DRANet augments as many images as the number of target images using one source image as shown in Fig. 4. DRANet-based data augmentation makes the classifier even more robust than the target-only model. Moreover, we show the content similarity matrix in Fig. 5 that reveals how well our model disentangles the representation into content and style components. We use 10 images with similar content from MNIST and MNIST-M each, and observe the confusion matrix has the highest diagonal values. We also observe that both higher values around 50% for both samples of digit one. The results show that our model disentangles the representation of content and style while maintaining each domain's characteristics.

4.2. Adaptation for Semantic Segmentation

To show the applicability of DRANet on the complex real-world scenario, we use GTA5 [29] and Cityscapes [5], which contain driving scene images with dense annotations. We train our model using 24966 images in GTA5 and 2975 images in Cityscapes train set, and we train DRN-26 [40] with 19 common classes for synthetic to real adaptation. The results in Fig. 6 show that our model generates stylized images following the artistic appearance of target images while keeping the scene structure of source images. We also evaluate the domain adaptation performance on semantic segmentation. The quantitative results in Tab. 2 show that our model achieves state-of-the-art performance

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**	**fwIoU**	**Pixel Acc.**
Source only	42.7	26.3	51.7	5.5	6.8	13.8	23.6	6.9	75.5	11.5	36.8	49.3	0.9	46.7	3.4	5.0	0.0	5.0	1.4	21.7	47.4	62.5
CyCADA [15]	79.1	33.1	77.9	23.4	17.3	32.1	**33.3**	**31.8**	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5	72.4	82.3
LC [39]	83.5	35.2	79.9	24.6	16.2	**32.8**	33.1	**31.8**	81.7	29.2	66.3	**63.0**	14.3	**81.8**	21.0	26.5	8.5	16.7	**24.0**	40.5	75.1	84.0
Ours (without CADT)	83.5	33.7	80.7	22.7	19.5	25.2	28.6	25.8	84.1	32.8	**84.4**	53.3	13.6	75.7	**21.7**	30.6	15.8	**20.3**	19.5	40.6	75.6	84.9
Ours (with CADT)	**85.0**	**35.8**	**82.0**	**26.4**	**21.6**	27.0	29.2	28.1	84.2	**34.0**	81.9	53.6	**15.9**	73.6	21.1	**31.0**	**16.7**	17.2	22.8	**41.4**	**76.4**	**85.7**
Target only	97.3	79.8	88.6	32.5	48.2	56.3	63.6	73.3	89.0	58.9	93.0	78.2	55.2	92.2	45.0	67.3	39.6	49.9	73.6	67.4	89.6	94.3

Table 2. Result comparison of DRANet to state-of-the-art methods on domain adaptation for semantic segmentation. We also report the performance of DRANet with and without Content-Adaptive Domain Transfer (CADT).

(a) Test images (CityScapes)

(b) Source prediction.

(c) Our prediction

(d) Ground Truth
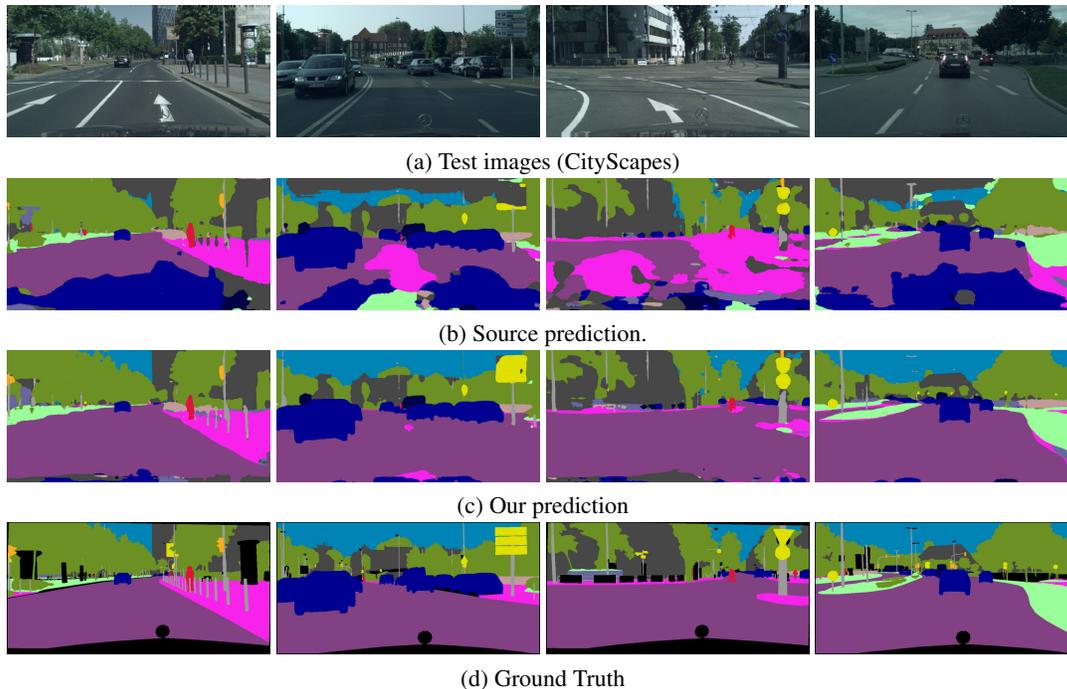
Figure 7. Semantic segmentation results for GTA5 to CityScapes. Note that we do not use any GT segmentation labels for training DRANet.

in all three main metrics for semantic segmentation: mIoU, fwIoU, and pixel accuracy. Among the 19 segmentation labels, our method outperforms the competitive methods in 14 categories. Especially, the accuracy of sky labels is improved by a large margin. We believe that our model designed for maintaining the scene structure allows to stably generate domain transferred images as shown in Fig. 6, and leads the performance improvement as shown in Fig. 7.

4.3. Ablation Study

Representation Disentangler   We design our separator incorporating two key ideas: one is the non-linearity of feature mapping and the other is domain normalization factor. To show the effectiveness of these key contributions, we set four experiment settings with/without non-linearity and normalization factors in our framework. We evaluate DRANet in each set for two bidirectional domain transfer tasks (one between MNIST and USPS, the other between MNIST and MNIST-M). We compare the classifi-

cation results of each case in unsupervised domain adaptation. As shown in Tab. 3, our model involving both non-linearity and normalization factors shows the best performance among four different settings. In the adaptation task between MNIST and MNIST-M, all model, even without non-linearity and normalization factor, produces the reasonable performance because both datasets contain the same content representation. Note that MNIST-M is one variation on MNIST proposed for unsupervised domain adaptation, which replaces the background of images while maintaining each MNIST digit [9]. However, there is a large gap in each case for adaptation between MNIST and USPS, which have obviously different content representation. The model without both components results in poor classification performance of one side. This means the model can only adapt either directional domain adaptation (MNIST to USPS or USPS to MNIST), like what the existing methods do. The model with either non-linearity or normalization improves the performance while our model with both factors achieves

(a) Source image (GTA5) (b) Ours: CADT using all image in (d) (c) DT using the rightmost image in (d)
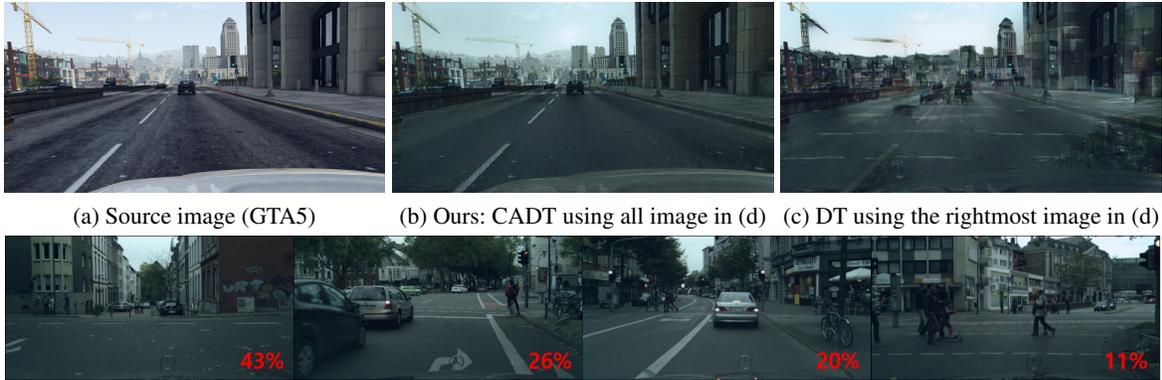
(d) Target images (CityScapes) in a minibatch and their content similarity with the source image in (a)

Figure 8. Comparison on image synthesis using Content-Adaptive Domain Transfer (CADT) and normal Domain Transfer (DT).

Non-linearity	Normal-ization	MNIST to USPS	USPS to MNIST	MNIST to MNIST-M	MNIST-M to MNIST
		11.2	87.1	97.0	99.0
	✓	90.7	90.2	97.7	99.1
✓		96.6	90.9	97.3	98.9
✓	✓	**98.2**	**97.8**	**98.7**	**99.3**

Table 3. Ablation study on the separator design to verify the effectiveness of the non-linearity in representation disentanglement (Non-linearity) and distribution scale parameters (Normalization).

the best among the other settings. We empirically demonstrate that non-linear mapping affords better representation disentanglement and the drastic performance improvement. As the advantages of non-linear mapping function of features proven in [31], we believe that the non-linearity is considerably responsible for clear separation of representations. We also show the normalization factor further boosts the adaptation performance. We can conclude that both factors play an important role in representation disentanglement as well as in an unsupervised domain adaptation.

Content-Adaptive Domain Transfer This subsection shows two advantages of our CADT for domain adaptation. One is that it prevents the model to be trained with bad training samples and the other is that it encourages the model to generate better-stylized images. During the early phase of training, the separator is not able to clearly disentangle the content and style components, which means each separation does not solely involve its identical information. Consequently, the model generates content-mixed images at the early training stage, and it might disturb the training by fooling discriminator, especially in the case the two images have quite different content. These strengths can be observed in Fig. 8 that shows the comparison of the results from the model with/without CADT trained with less than 1000 iterations. Fig. 8-(a) is a source image (GTA5), and Fig. 8-(d) contains multiple target images (Cityscapes) in one minibatch. The bottom-right digit indicates the content similarity with source image. We show the domain transferred images with/without CADT in Fig. 8-(b),(c), respec-

tively. The result in Fig. 8-(c) is generated by adapting the domain of rightmost target image in Fig. 8-(d), which has the lowest similarity. The results show the normal domain transfer causes the significant artifact in the early stage of training while the proposed CADT reasonably synthesizes the image. It means that CADT helps to disentangle the representation even at just a few iterations. We also show that the general performance improvement by CADT in Tab. 2 by comparing the domain adaptation results with/without CADT. The table demonstrates the effectiveness of our content-adaptive domain transfer.

5. Conclusion

In this paper, we present a new network architecture called DRANet which disentangles individual feature representations into two factors, content and style, and transfers domains by applying the style features of another domain. In contrast to conventional methods which focus on the associations of features among domains, we learn the distinctive features of each domain, then separate the features into two components. This design enables us to transfer the domains multi-directionally with our single model. In addition, our method does not require any class labels for adapting domains. Another contribution of this work is to propose a content-adaptive domain transfer method to synthesize more realistic images from the complex scene structures. Extensive experiments show that our model synthesizes visually pleasing images transferred across domains, and the synthesized images boost the performance of the classification and semantic segmentation tasks. We also demonstrate that the proposed method outperforms the state-of-the-art domain adaptation methods despite the absence of any labeled data for training.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems (NIPS)*, 19:137–144, 2006.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3722–3731, 2017.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 343–351, 2016.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2172–2180, 2016.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1 (Long and Short Papers), page 4171–4186, 2018.

[8] Ahmed Elgammal and Chan-Su Lee. Separating style and content on a nonlinear manifold. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 478–485. IEEE, 2004.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[10] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 262–270, 2015.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[12] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1287–1298, 2018.

[13] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. 1(2), 2016.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1989–1998. PMLR, 2018.

[16] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[21] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 469–477, 2016.

[22] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8867–8876, 2018.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 97–105. PMLR, 2015.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2018.

[26] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *International Conference on Learning Representations (ICLR)*, 2018.

[27] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2558–2567, 2018.

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2016.

[29] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 102–118. Springer, 2016.

[30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2234–2242, 2016.

[31] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[32] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 443–450. Springer, 2016.

[33] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000.

[34] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.

[35] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2672–2681, 2019.

[36] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015.

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017.

[38] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[39] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: a separable component for unsupervised domain adaptation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13736–13745, 2020.

[40] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[41] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 7354–7363. PMLR, 2019.

[42] Rui Zhang, Sheng Tang, Yu Li, Junbo Guo, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Style separation and synthesis via generative adversarial networks. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 183–191, 2018.

[43] Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8447–8455, 2018.

[44] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017.

[45] Yang Zou, Xiaodong Yang, Zhiding Yu, BVK Kumar, and Jan Kautz. Joint disentangling and adaptation for cross-domain person re-identification. *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.