Pixel-Level Domain Transfer

Donggeun Yoo[1] dgyoo@rcv.kaist.ac.kr
Namil Kim[1] nikim@rcv.kaist.ac.kr
Sunggyun Park[1] sunggyun@kaist.ac.kr
Anthony S. Paek[2] apaek@lunit.io
In So Kweon[1] iskweon@kaist.ac.kr

[1]KAIST [2]Lunit Inc.

Abstract. We present an image-conditional image generation model. The model transfers an input domain to a target domain in semantic level, and generates the target image in pixel level. To generate realistic target images, we employ the real/fake-discriminator in Generative Adversarial Nets [1], but also introduce a novel domain-discriminator to make the generated image relevant to the input image. We verify our model through a challenging task of generating a piece of clothing from an input image of a dressed person. We present a high quality clothing dataset containing the two domains, and succeed in demonstrating decent results.

Keywords: Domain transfer, Image generation, Generative Adversarial Network.

1 Introduction

Every morning, we agonize in front of the closet over what to wear, how to dress up, and imagine ourselves with different clothes on. To generate mental images [2] of ourselves wearing clothes on a hanger is an effortless work for our brain. In our daily lives, we ceaselessly perceive visual scene or objects, and often transfer them to different forms by the mental imagery. Our focus of this paper lies on the problem; to enable a machine to transfer a visual input into different forms and to visualize the various forms by generating a pixel-level image.

Image generation has been attempted by a long line of works [3,4,5] but generating realistic images has been challenging since an image itself is high dimensional and has complex relations between pixels. However, recently a few machines have succeeded in generating realistic images [1,6,7,8], with the drastic advances of deep learning. Although these works are similar to ours in terms of image generation, ours is distinct in terms of *image-conditioned image generation.* We take an image as a conditioned input lying on a domain, and re-draw a target image lying on another.

In this work, we define two domains; a source domain and a target domain. The gap between the two is connected by a semantic meaning. For instance, if we define an image of a dressed person as a source domain, a piece of the person's clothing is defined as the target domain. Transferring an image domain into a

A source image. Possible target images.

Fig. 1. A real example showing non-deterministic property of target image in the pixel-level domain transfer problem.

different image domain has been proposed in computer vision [9,10,11,12,13,14], but all these adaptations take place in the feature space, i.e. the model parameters are adapted. However, our method directly produces target images.

We transfer a source domain to a pixel-level target while overcoming the semantic gap between domains. Transferred image should look realistic yet preserving the semantic meaning. To do so, we present a pixel-level domain converter composed of an encoder for semantic embedding of a source and a decoder to produce a target image. However, training the converter is not straightforward because the target is not deterministic [15]. Given a source image, one can generate infinite targets satisfying the semantic meaning, by shifting and scaling objects, changing illumination, or changing object shapes as the examples in Fig. 1 show. To challenge this problem, we introduce two strategies as follows.

To train our converter, we first place a separate network named *domain discriminator* on top of the converter. The domain discriminator takes a pair of a source image and a target image, and is trained to make a binary decision whether the input pair is associated or not. The domain discriminator then supervises the converter to produce associated images. Both of the networks are jointly optimized by the adversarial training method, which Goodfellow *et al.* [1] propose for generating realistic images. Such binary supervision solves the problem of non-deterministic property between the domains and enables us to train the semantic connection between the domains. Secondly, in addition to the domain discriminator, we also employ the discriminator of [1], which is supervised by the labels of "real" or "fake", to produce realistic images.

Our framework deals with the three networks that play distinct roles. Labels are given to the two discriminators, and they supervise the converter to produce images that are realistic yet keeping the semantic meaning. Those two discriminators become unnecessary after the training stage and the converter is our ultimate goal. We verify our method by quite challenging settings; the source domain is a natural human image and the target domain is a product image of the person's top. To do so, we have made a large dataset named LookBook, which contains in total of 84k images, where 75k human images are associated with 10k top product images. With this dataset, our model succeeds in generating moderate target images, and the evaluation result verifies the effectiveness of our *domain discriminator* to train the converter.

Contributions In summary, our contributions are,

1. Proposing the first framework for semantically transferring a source domain to a target domain in pixel-level.
2. Proposing a novel discriminator that enables us to train the semantic relation between the domains.
3. Building a large clothing dataset containing two domains, which is expected to contribute to a wide range of domain adaptation researches.

2 Related Work

Our work is highly related with the image-generative models since our final result from an input image is also an image. The image-generative models can be grouped into two families; generative parametric approaches [3,4,5] and adversarial approaches [1,16,17,18]. The former approaches generate images from latent representation by a generative decoder but often have troubles in training complexities, which results in a low rate of success in generating realistic natural images. The adversarial approaches originate from Generative Adversarial Nets (GAN) proposed by Goodfellow *et al.* [1]. GAN framework introduces a generator (i.e. a decoder), which captures the data distribution so that can generate images, and a discriminator, which distinguishes between generated samples and real images. The two networks are optimized to go against each other; the discriminator is trained to distinguish between real and fake samples while the generator is trained to confuse the discriminator. Mirza and Osindero [16] extend GAN to a class conditional version, and Denton *et al.* [17] improve the image resolution in a coarse-to-fine fashion. However, GAN is known to be unstable due to the adversarial training, often resulting in incomprehensible or noisy images. Quite recently, Radford *et al.* [18] have proposed architectures named Deep Convolutional GANs, which is stable to be trained, and have succeeded in generating high quality images. Besides the two families, a recurrent network based model [6] and a deconvolutional network based model [19] have also been proposed.

The recent improvements of GAN framework and its successful results motivate us to adopt the networks. We replace the generator with our converter which is an image-conditioned model, while [16] is class-conditional and [15] is attribute-conditional. The generator of Mathieu *et al.* [20] is similar to ours in that it is conditioned with video frames to produce next frames. They add a mean square loss to the generator to strongly relate the input frames to the next frames. However, we cannot use such loss due to the non-deterministic property of the target domain. We therefore introduce a novel discriminator named domain discriminator.

Our work is also related with transfer learning, also called as domain adaptation, in terms of transferring the domain. This aims to transfer the model parameter trained on a source domain to a different domain. For visual recognition, many methods to adapt domains [9,10,11] have been proposed. Especially for the recent use of the deep convolutional neural network [21], it has been common to pre-train a large network [22] over ImageNet [23] and transfer the parameters to a target domain [12,24,25]. Similar to our clothing domains, Chen *et al.* [13] and Huang *et al.* [14] address a gap between fashion shopping mall images and unconstrained human images for the clothing attribute recognition [13] and the product retrieval [14]. However, all these methods are different from ours in respect of cross-domain image *generation.* The adaptation of these related works takes place in the feature space, while we directly produce target images from the source images.

3 Review of Generative Adversarial Network

Generative Adversarial Nets (GAN) [1] is a generalized framework for generative models which [17,18,20] and we utilize for visual data. In this section, we briefly review GAN in the context of image data. GAN is framed by an adversarial setting of two networks; a generator and a discriminator. The eventual goal of the generator is to map a small dimensional space Z to a continuous pixel-level image space, i.e., to enable the generator to produce a realistic image from an input random vector $z \in Z$.

To train such a generator, a discriminator is introduced. The discriminator takes either a real image or a fake image drawn by the generator, and distinguish whether its input is real or fake. The training procedure can be intuitively described as follows. Given an initialized fool generator G^0, an initial discriminator D_R^0 is firstly trained with real training images $\{I^i\}$ and fake images $\{\hat{I}^j = G^0(z^j)\}$ drawn by the generator. After that, we freeze the updated discriminator D_R^1 and train the generator G^0 to produce better images, which would lead the discriminator D_R^1 to misjudge as real images. These two procedures are repeated until they converge. The training can be represented as a minimax objective as,

$$\min_{\Theta^G} \max_{\Theta_R^D} \mathbb{E}_{I \sim p_{\text{data}}(\mathbf{I})}[\log(D_R(I))] + \mathbb{E}_{z \sim p_{\text{noise}}(\mathbf{z})}[\log(1 - D_R(\hat{I}))], \tag{1}$$

where Θ^G and Θ_R^D indicate the model parameters of the discriminator and the generator respectively. Here, the discriminator produces a scalar probability that is high when the input I is real but otherwise low. The discriminator loss function $\mathcal{L}_R^D$ is defined as the binary cross entropy,

$$\mathcal{L}_R^D(I) = -t \cdot \log[D_R(I)] + (t-1) \cdot \log[1 - D_R(I)], \qquad \text{s.t.} \;\; t = \begin{cases} 1 & \text{if } I \in \{I^i\} \\ 0 & \text{if } I \in \{\hat{I}^j\}. \end{cases} \tag{2}$$

One impressive fact in the GAN framework is the lowest level of supervision; real or fake. Without strong and fine supervisions (e.g. mean square error between images), this framework succeeds in generating realistic images. This motivates us to raise the following question. Under such a low-level supervision, would it be possible to train a connection between distinct image domains? If so, could we transform an image lying on a domain to a realistic image lying on another? Through this study, we have succeeded in doing so, and the method is to be presented in Sec. 4.

4 Pixel-Level Domain Transfer

In this section, we introduce the pixel-level domain transfer problem. Let us define a source image domain $S \subset \mathbb{R}^{W \times H \times 3}$ and a target image domain $T \subset \mathbb{R}^{W \times H \times 3}$. Given a transfer function named converter C, our task is to transfer a source image $I_S \in S$ to a target image $\hat{I}_T \in T$ such as

$$\hat{I}_T = C(I_S | \Theta^C), \tag{3}$$

where Θ^C is the model parameter of the converter. Note, again, that the inference $\hat{I}_T$ is not a feature vector but itself a target image of $W \times H \times 3$ size. To do so, we employ a convolutional network model for the converter C, and adopt a supervised learning to optimize the model parameter Θ^C. In the training data, each source image I_S should be associated with a ground-truth target image I_T.

4.1 Converter Network

Our target output is *pixel-level* image. Furthermore, the two domains are related with a *semantic* meaning. Pixel-level output production itself is a challenging problem but the semantic transfer makes the problem even more difficult. A converter should be designed to selectively summarize the semantic attributes from a source image and then produce a transformed pixel-level image.

The top network in Fig. 2 shows the architecture of the converter we propose. The converter is a unified network that is end-to-end trained but we can divide it into two parts according to their counter roles; an encoder and a decoder. The encoder part is composed of five convolutional layers to abstract the source into a semantic 64-dimensional code. This abstraction procedure is significant since our source domain (e.g. natural fashion image) and target domain (e.g. product image) are paired in a semantic content (e.g. the product). The 64-dimensional code should capture the semantic attributes (e.g. category, color, etc.) of a source to be well decoded into a target. The code is then fed by the decoder, which constructs a relevant target through the five decoding layers. Each decoding layer conducts the fractional-strided convolutions, where the convolution operates in the opposite direction. The reader is referred to Table 1 for more details about the architectures of the encoder and the decoder.

4.2 Discriminator Networks

Given the converter, a simple choice of a loss function to train it is the mean-square error (MSE) such as $||\hat{I}_T - I_T||_2^2$. However, MSE may not be a proper choice due to critical mismatches between MSE and our problem. Firstly, MSE is

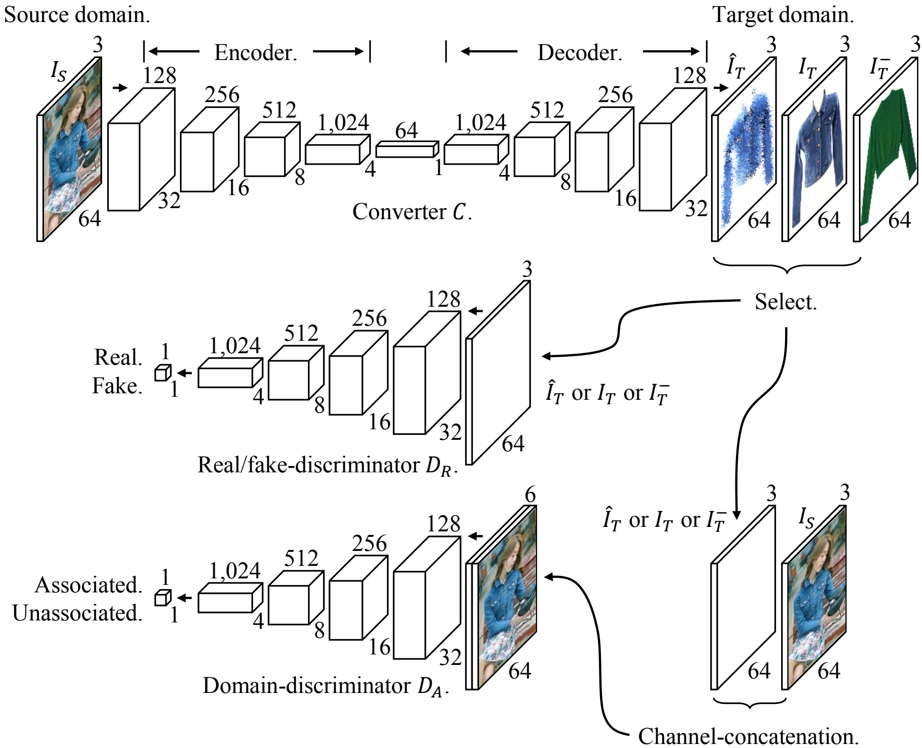

Fig. 2. Whole architecture for pixel-level domain transfer.

not suitable for pixel-level supervision for natural images. It has been well known that MSE is prone to produce blurry images because it inherently assumes that the pixels are drawn from Gaussian distribution [20]. Pixels in natural images are actually drawn from complex multi-modal distributions. Besides its intrinsic limitation, it causes another critical problem especially for the pixel-level domain transfer as follows.

Given a source image, the target is actually not unique in our problem. Our target domain is the lowest pixel-level image space, not the high-level semantic feature space. Thus, the number of possible targets from a source is infinite. Fig. 1 is a typical example showing that the target is not unique. The clothing in the target domain is captured in various shapes, and all of the targets are true. Besides the shapes, the target image can be captured from various viewpoints, which results in geometric transformations. However, minimizing MSE always forces the converter to fit into one of them. Image-to-image training with MSE never allows a small geometric mis-alignment as well as various shapes. Thus, training the converter with MSE is not a proper use for this problem. It would be better to introduce a new loss function which is tolerant to the diversity property of the pixel-level target domain.

Layer	Number of filters	Filter size (w×h×ch)	Stride	Pad	Batch norm.	Activation function
Conv. 1	128	5×5×{3, 3, 6}	2	2	×	L-ReLU
Conv. 2	256	5×5×128	2	2	○	L-ReLU
Conv. 3	512	5×5×256	2	2	○	L-ReLU
Conv. 4	1,024	5×5×512	2	2	○	L-ReLU
Conv. 5	{64, 1, 1}	1×1×1,024	1	0	{○, ×, ×}	{L-ReLU, sigmoid, sigmoid}

(a) Details of the {encoder, real/fake discriminator, domain discriminator}.

Layer	Number of filters	Filter size (w×h×ch)	Stride	Pad	Batch norm.	Activation function
Conv. 1	4×4×1,024	1×1×64	1	0	○	ReLU
F-Conv. 2	1,024	5×5×512	1/2	-	○	ReLU
F-Conv. 3	512	5×5×256	1/2	-	○	ReLU
F-Conv. 4	256	5×5×128	1/2	-	○	ReLU
F-Conv. 5	128	5×5×3	1/2	-	×	tanh

(b) Details of the decoder.

Table 1. Details of each network. In (a), each entry in $\{\cdot\}$ corresponds to each network. L-ReLU is leaky-ReLU. In (b), F denotes fractional-stride. The activation from the first layer is reshaped into 4×4×1,024 size before being fed to the second layer.

In this paper, on top of the converter, we place a discriminator network playing a role as a loss function beyond MSE. As in [1,17,18], the discriminator network guides the converter to produce realistic target under the supervision of real/fake. However, this is not the only role that our discriminator plays. If we simply use the original discriminator replacing MSE, a produced target could look realistic but its contents may not be relevant to the source. This is because there is no pair wise supervision such as MSE. Only real/fake supervision exists.

Given arbitrary image triplets $(I_S^+, I_S^\oplus, I_S^-)$ in the source domain S, where I_S^+ and $I_S^\oplus$ are about the same object while I_S^- is not, a converter transfers them into the images $(\hat{I}_T^+, \hat{I}_T^\oplus, \hat{I}_T^-)$ in the target domain T. Let us assume that these transfered images look realistic enough due to the real/fake discriminator. Beyond the realistic results, the best converter C should satisfy the following condition,

$$s\left(\hat{I}_T^+, \hat{I}_T^\oplus\right) > s\left(\hat{I}_T^+, \hat{I}_T^-\right) \text{ and } s\left(\hat{I}_T^+, \hat{I}_T^\oplus\right) > s\left(\hat{I}_T^\oplus, \hat{I}_T^-\right), \tag{4}$$

where $s(\cdot)$ is a semantic similarity function. This condition means that an estimated target should be semantically associated with the source. One supervision candidate to let the converter C meet the condition is the combined use of MSE with the real/fake loss, just as what [20] does for the video prediction. However, again, it is not the best option for our problem because the ground-truth I_T is not unique. Thus, we propose a novel discriminator, named domain discriminator, to take the pair wise supervision into consideration.

The domain discriminator D_A is the lowest network illustrated in Fig. 2. To enable pair wise supervision while being tolerant to the target diversity, we significantly loosen the level of supervision compared to MSE. The network D_A takes a pair of source and target as input, and produces a scalar probability of whether the input pair is associated or not. Let us assume that we have a source I_S, its ground truth target I_T and an irrelevant target I_T^-. We also have an inference $\hat{I}_T$ from the converter C. We then define the loss $\mathcal{L}_A^D$ of the domain discriminator D_A as,

$$\mathcal{L}_A^D(I_S, I) = -t \cdot \log[D_A(I_S, I)] + (t-1) \cdot \log[1 - D_A(I_S, I)],$$
$$\text{s.t. } t = \begin{cases} t = 1 & \text{if } I = I_T \\ t = 0 & \text{if } I = \hat{I}_T \\ t = 0 & \text{if } I = I_T^-. \end{cases} \quad (5)$$

The source I_S is always fed by the network as one of the input pair while the other I is chosen among (I_T^-, $\hat{I}_T$, I_T) with equal probability. Only when the source I_S and its ground-truth I_T is paired as input, the domain discriminator is trained to produce high probability whereas it minimizes the probability in other cases. Here, let us pay more attention to the input case of (I_S, $\hat{I}_T$).

The produced target $\hat{I}_T$ comes from the source but we regard it as an unassociated pair (t=0) when we train the domain discriminator. Our intention of doing so is for *adversarial training* of the converter and the domain discriminator. The domain discriminator loss is minimized for training the domain discriminator while it is maximized for training the converter. The better the domain discriminator distinguishes a ground-truth I_T and an inference $\hat{I}_T$, the better the converter transfers the source into a relevant target.

In summary, we employ both of the real/fake discriminator and the domain discriminator for adversarial training. These two networks play a role as a loss to optimize the converter, but have different objectives. The real/fake discriminator penalizes an unrealistic target while the domain discriminator penalizes a target being irrelevant to a source. The architecture of the real/fake discriminator is identical to that of [18] as illustrated in Fig. 2. The domain discriminator also has the same architecture except for the input filter size since our input pair is stacked across the channel axis. Several architecture families have been proposed to feed a pair of images to compare them but a simple stack across the channel axis has shown the best performance as studied in [26]. The reader is referred to Table 1 for more details about the discriminator architectures.

4.3 Adversarial Training

In this section, we present how we iteratively train the converter C, the real/fake discriminator D_R and the domain discriminator D_A. We place the two discriminators on top of the converter, so the two loss functions have been defined. The real/fake discriminator loss $\mathcal{L}_R^D$ is Eq. (2), and the domain discriminator loss $\mathcal{L}_A^D$ is Eq. (5). With these two losses, we basically follow the adversarial training procedure of [1], as explained in Sec. 3.

Algorithm 1: Adversarial training for the pixel-level domain transfer.

Set the learning rate η and the batch size B.
Initialize each network parameters $\Theta^C, \Theta^D_R, \Theta^D_A$,
Data: Paired image set $\{I^n_S, I^n_T\}^N_{n=1}$.
while *not converged* **do**
  Get a source batch $\left\{I^i_S\right\}^B_{i=1}$ and a target batch $\left\{I^i\right\}^B_{i=1}$,
    where I^i is a target sample randomly chosen from $(I^i_T, I^{i-}_T, \hat{I}^i_T)$.
  Update the real/fake discriminator D_R:
    $\Theta^D_R \leftarrow \Theta^D_R - \eta \cdot \frac{1}{B}\sum^B_{i=1} \frac{\partial \mathcal{L}^D_R(I^i)}{\partial \Theta^D_R}$
  Update the domain discriminator D_A:
    $\Theta^D_A \leftarrow \Theta^D_A - \eta \cdot \frac{1}{B}\sum^B_{i=1} \frac{\partial \mathcal{L}^D_A(I^i_S, I^i)}{\partial \Theta^D_A}$
  Update the converter C:
    $\Theta^C \leftarrow \Theta^C - \eta \cdot \frac{1}{B}\sum^B_{i=1} \frac{\partial \mathcal{L}^C(I^i_S, I^i)}{\partial \Theta^C}$
end

Given a paired image set for training, let us assume that we get a source batch $\{I^i_S\}$ and a target batch $\{I^i\}$ where a target sample I^i is stochastically chosen from $(I^i_T, I^{i-}_T, \hat{I}^i_T)$ with equal probability. At first, we train the discriminators. We train the real/fake discriminator D_R with the target batch to reduce the loss of Eq. (2). The domain discriminator D_A is trained with both of source and target batches to reduce the loss of Eq. (5). After that, we freeze the updated discriminator parameters $\{\hat{\Theta}^D_R, \hat{\Theta}^D_A\}$, and optimize the converter parameters Θ^C to *increase* the losses of both discriminators. The loss function of the converter can be represented as,

$$\mathcal{L}^C(I_S, I) = -\frac{1}{2}\mathcal{L}^D_R(I) - \frac{1}{2}\mathcal{L}^D_A(I_S, I), \text{ s.t. } I = \text{sel}\left(\{I_T, \hat{I}_T, I^-_T\}\right), \quad (6)$$

where sel$(\cdot)$ is a random selection function with equal probability. The reader is refered to Algorithm 1 for more concrete training procedures.

5 Evaluation

In this section, we verify our pixel-level domain transfer method by undergoing a challenging task; a natural human image belongs to the source domain, and a product image of that person's top belongs to the target domain. We first give description on the dataset we collect and use for training in Sec. 5.1. We then provide details on the experimental setting in Sec. 5.2, and we demonstrate and discuss the results and in Sec. 5.3.

5.1 LookBook Dataset

We make a dataset named LookBook that covers two fashion domains. Images of one domain contain fashion models, and those of the other domain contains

Fig. 3. Example images of LookBook. A product image is associated with multiple fashion model images.

top products with a clean background. Real examples are shown in Fig. 3. We manually associate each product image with corresponding images of a fashion model fitting the product, so each pair is accurately connected with the same product. LookBook contains 84,748 images where 9,732 top product images are associated with 75,016 fashion model images. It means that a product has around 8 fashion model images in average. We collect the images from five on-line fashion shopping malls[1] where a product image and its fashion model images are provided. Although we utilize LookBook for the pixel-level domain transfer, we believe that it can contribute to a wide range of domain adaptation researches.

Chen *et al.* [13] also has presented a similar fashion dataset dealing with two domains. However, it does not fit with our task since the domains are differently defined in details. They separate the domain into user taken images and on-line shopping mall images so that both domains include humans.

5.2 Experiment Details

Data preparation Before training, we rescale all images in LookBook to have 64 pixels at a longer side while keeping the aspect ratio, and fill the margins of both ends with 255s. Pixel intensities are also normalized to a range of $[-1, 1]$ according to the tanh activation layer of the converter. We then randomly select 5% images to define a validation set, and also 5% images for a test set. Since LookBook has 9,732 products, each of the validation set and the test set is composed of 487 product images and their all fashion model images. The remaining images compose a training set.

Training The filters of the three networks are randomly initialized from a zero mean Gaussian distribution with a standard deviation of 0.02. The leak slope of the LeakyReLU in Table 1-(a) is 0.2. All models were trained with stochastic gradient descent (SGD) with mini-batch of 128 size. We also follow the learning rate of 0.0002 and the momentum term of 0.5 suggested by Radford *et al.* [18] to stabilize the training. After 25 epochs, we lessen the learning rate to 0.00002 to finely tune the parameter for 5 more epochs.

[1] {bongjashop, jogunshop, stylenanda}.com, {smallman, wonderplace}.co.kr

Baselines We compare our method with two baselines. The first baseline is a naïve converter trained only with the mean square loss, noted by "Converter+MSE". The second baseline is a converter trained with the real/fake discriminator but without the domain discriminator for the purpose of showing the impact of our domain discriminator. This is noted by "Converter+RF-Discrim". The training details of the two baselines are identical to those of ours.

User study Due to the semantic gap between domains, the target domain is not deterministic, i.e. no unique ground-truth exists. It makes the quantitative analysis difficult. Thus, we conduct a user study on our test results as a primary evaluation. For this study, we created a sub-test set composed of 100 source images randomly chosen from the test set. For each source image, we showed users three target images generated by the two baselines and our method. Users were asked to rate them three times in accordance with three different evaluation criteria as follows. A total of 25 users participated in this study.

1. How realistic is each result? Give a score from 0 to 2.
2. How well does each result capture the attributes (color, texture, logos, etc.) of the source image? Give a score from 0 to 2.
3. Is the category of each result identical to that of the source image? Give a binary score of 0 or 1.

Pixel-level dissimilarity For secondary quantitative analysis, we measure a pixel-level dissimilarity by computing the root mean square error between a generated image and a target image over the test set. However, this measure is advantageous for the blurry results of "Converter+MSE" method because it is trained by minimizing the mean square loss. To compensate the effect, as suggested in [15], we also measure the root mean square error in another setting where the "Converter+RF-Discrim." results and ours are blurred with an average filter of 10×10 size.

5.3 Results and Analysis

In this section, we demonstrate and discuss the experimental results. First, we show qualitative results in Fig. 4, where the examples are chosen from the test set. The images generated by the proposed method look more relevant to the source image and more realistic compared to those of baselines. Boundaries of products are sharp, and small details such as stripes, patterns are well described in general. The results of "Converter+RF-Discim" look realistic but irrelevant to the source image, and those of "Converter+MSE" is quite blurry. Let us discuss and compare the results with quantitative evaluation.

Fig. 4. Qualitative comparison. Each image from the left to the right respectively corresponds to a source image, "Converter+RF-Discrim." result, "Converter+MSE" result and our result. Each image is shown in 64×64×3 dimensions.

Method	User study score			Pixel-level dissimilarity	
	Realistic	Attribute	Category	RMSE	RMSE-Blur
Converter+RF-Discrim.	0.4013	0.2100	0.0620	98.58	91.91
Converter+MSE	0.2771	0.5974	0.5966	**65.56**	**65.56**
Ours	**0.8210**	**0.6734**	**0.7645**	80.75	70.62

Table 2. Quantitative evaluation through a user study, and measuring root mean square error between a generated target and a ground-truth. The user-study scores are normalized to a range of $[0, 1]$.

Table 2 shows the quantitative evaluation results. The scores are averaged and normalized to a range of $[0, 1]$. In the "Realistic" criteria, it is not surprising that "Converter+MSE" shows the worst performance due to the intrinsic limitation of the mean square loss for image generation. Its assumption of Gaussian distribution results in producing quite blurry as shown in Fig. 4. However, the strong pair wise supervision of the mean square loss relatively succeeds in representing the category and attributes of a product.

When the converter is supervised with the real/fake domain discriminator only, the generated images are more realistic than those of "Converter+MSE". However, it fails to connect between the source and the target domains and yields much lower attribute and category scores than those of "Converter+MSE". As shown in Fig. 4, "Converter+RF-Discrim." mostly produces irrelevant targets.

The user study results demonstrate the effectiveness of the proposed method. For all valuation criteria, our method clearly outperforms the baselines. Especially, the ability to capture attributes and categories of products is better than that of "Converter+MSE" although our converter is trained under the low-level supervision of "associated" or "unassociated". This result verifies the effectiveness of our domain discriminator. In Fig. 4, compared to the baselines, our results well capture the category and attribute of products while being realistic.

Another interesting observation is that our score of "Realistic" criteria is higher than that of "Converter+RF-Discrim.". Both of the methods include the real/fake discriminator but demonstrate distinct results. The difference may be caused by the domain discriminator which is added to the adversarial training in our method. When we train the domain discriminator, we regard all produced targets as "unassociated". This setting makes the the converter better transfer a source image into a more *realistic* and relevant target.

We also measure a pixel-level dissimilarity by computing a root mean square error between a generated image and a ground-truth. As we can expect, "Converter+MSE" shows the best result because the converter is trained by minimizing the mean square loss. However, the dissimilarity of our method is decreased and closer to that of "Converter+MSE" when we add the blur effect to our result images with an average filter of 10×10 size.

Fig. 5 verifies how well the encoder of the converter encodes clothing attributes against to varying conditions of source images. The source images sig-

Fig. 5. Generation results according to varying input conditions. The odd rows are source images, and the even rows are generation results. Each image is shown in 64×64×3 dimensions.

nificantly vary in terms of backgrounds, viewpoints, human poses and self-occlusions. Despite these variations, our converter generates less varying targets while reflecting clothing attributes of source images. These results imply that the encoder is less influenced by varying conditions and robustly summarizes the source information in a semantic level.

Finally, we show results of the *inverse setting of domains* in Fig. 6. Since generating fashion models is a more complex task, we found that 65 epochs for initial training and 5 more epochs for fine-tuning are required in our experiment. All the other details are exactly identical to those of the original setting.

6 Conclusion

We have presented pixel-level domain transfer based on Generative Adversarial Nets framework. The proposed domain discriminator enables us to train the semantic relation between the domains, and the converter has succeeded in generating decent target images. Also, we have presented a large dataset that could contribute to further domain adaptation researches, by building a large clothing dataset which contains two domains.

Since our framework is not constrained for specific problems, we expect to extend it to other types of pixel-level domain transfer from low-level image processing to high-level synthesis. Furthermore, we would like to take some of the noticed challenges, including an unstable oscillation of training, inaccurate and wobble details, into consideration in our future work.

Fig. 6. 100 chosen results from the *inverse setting of domains*. Each image is shown in 64×64×3 dimensions.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. (2014) 2672–2680
2. Eysenck, M.W.: Fundamentals of cognition. Psychology Press East Sussex, UK, USA and Canada (2006)
3. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (2006) 504–507
4. Salakhutdinov, R., Hinton, G.E.: Deep boltzmann machines. In: International conference on artificial intelligence and statistics. (2009) 448–455
5. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 1096–1103
6. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: Proceedings of The 32nd International Conference on Machine Learning. (2015) 1462–1471
7. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of The 32nd International Conference on Machine Learning. (2015) 2256–2265
8. Theis, L., Bethge, M.: Generative image modeling using spatial lstms. In: Advances in Neural Information Processing Systems. (2015) 1918–1926
9. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Computer Vision–ECCV 2010. Springer (2010) 213–226
10. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1785–1792
11. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 999–1006
12. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1717–1724
13. Chen, Q., Huang, J., Feris, R., Brown, L.M., Dong, J., Yan, S.: Deep domain adaptation for describing people based on fine-grained clothing attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5315–5324
14. Huang, J., Feris, R.S., Chen, Q., Yan, S.: Cross-domain image retrieval with a dual attribute-aware ranking network. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1062–1070
15. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. arXiv preprint arXiv:1512.00570 (2015)
16. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
17. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems. (2015) 1486–1494
18. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

19. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1538–1546
20. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)
21. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4) (1989) 541–551
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3) (2015) 211–252
24. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2014) 806–813
25. Yoo, D., Park, S., Lee, J.Y., Kweon, I.: Multi-scale pyramid pooling for deep convolutional representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2015) 71–80
26. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4353–4361