Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

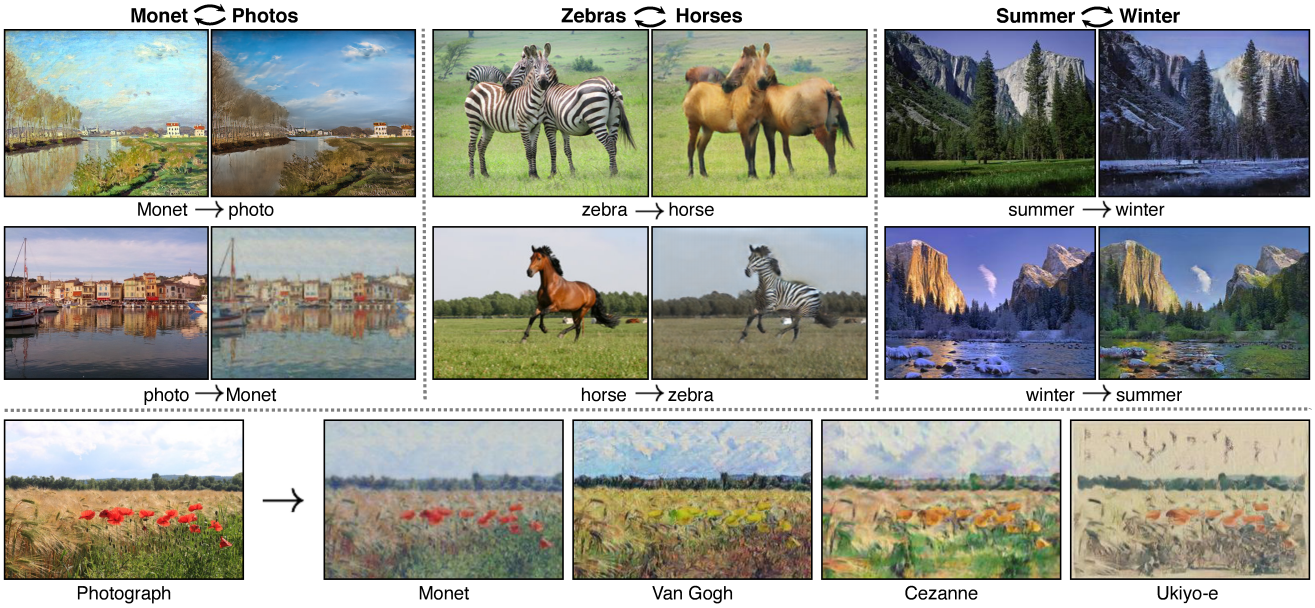

Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa. Example application (*bottom*): using a collection of paintings of a famous artist, learn to render a user's photograph into their style.

Abstract

Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs. However, for many tasks, paired training data will not be available. We present an approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples. Our goal is to learn a mapping $G : X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, we couple it with an inverse mapping $F : Y \rightarrow X$ and introduce a cycle consistency loss *to push $F(G(X)) \approx X$ (and vice versa). Qualitative results are presented on several tasks where paired training data does not exist, including collection style transfer, object transfiguration, season transfer, photo enhancement, etc. Quantitative comparisons against several prior methods demonstrate the superiority of our approach.*

1. Introduction

What did Claude Monet see as he placed his easel by the bank of the Seine near Argenteuil on a lovely spring day in 1873 (Figure 1, top-left)? A color photograph, had it been invented, may have documented a crisp blue sky and a glassy river reflecting it. Monet conveyed his *impression* of this same scene through wispy brush strokes and a bright palette. What if Monet had happened upon the little harbor in Cassis on a cool summer evening (Figure 1, bottom-left)? A brief stroll through a gallery of Monet paintings makes it easy to imagine how he would have rendered the scene: perhaps in pastel shades, with abrupt dabs of paint, and a somewhat flattened dynamic range.

We can imagine all this despite never having seen a side by side example of a Monet painting next to a photo of the scene he painted. Instead we have knowledge of the set of Monet paintings and of the set of landscape photographs. We can reason about the stylistic differences between these two sets, and thereby imagine what a scene might look like if we were to "translate" it from one set into the other.

* indicates equal contribution

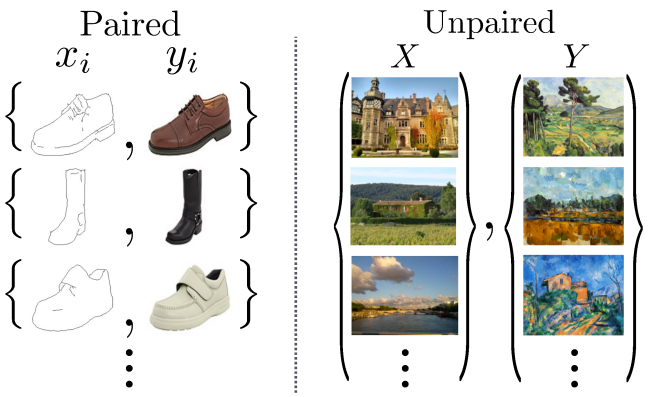

Figure 2: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^N$, where the y_i that corresponds to each x_i is given [20]. We instead consider *unpaired* training data (right), consisting of a source set $\{x_i\}_{i=1}^N \in X$ and a target set $\{y_j\}_{j=1}^M \in Y$, with no information provided as to which x_i matches which y_j.

In this paper, we present a system that can learn to do the same: capturing special characteristics of one image collection and figuring out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples.

This problem can be more broadly described as image-to-image translation [20], converting an image from one representation of a given scene, x, to another, y, e.g., grayscale to color, image to semantic labels, edge-map to photograph. Years of research in computer vision, image processing, and graphics have produced powerful translation systems in the supervised setting, where example image pairs $\{x, y\}$ are available (Figure 2, left), e.g., [9, 17, 20, 21, 24, 29, 41, 52, 54, 57]. However, obtaining paired training data can be difficult and expensive. For example, only a couple of datasets exist for tasks like semantic segmentation (e.g., [4]), and they are relatively small. Obtaining input-output pairs for graphics tasks like artistic stylization can be even more difficult since the desired output is highly complex, typically requiring artistic authoring. For many tasks, like object transfiguration (e.g., zebra→horse, Figure 1 top-middle), the desired output is not even well-defined.

We therefore seek an algorithm that can learn to translate between domains without paired input-output examples (Figure 2, right). We assume there is some underlying relationship between the domains – for example, that they are two different renderings of the same underlying world – and seek to learn that relationship. Although we lack supervision in the form of paired examples, we can exploit supervision at the level of sets: we are given one set of images in domain X and a different set in domain Y. We may train a mapping $G : X \rightarrow Y$ such that the output $\hat{y} = G(x)$, $x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify $\hat{y}$ apart from y. In theory, this objective can induce an output distribution over $\hat{y}$ that matches the empirical distribution $p_Y(y)$ (in general, this requires that G be stochastic) [14]. The optimal G thereby translates the domain X to a domain $\hat{Y}$ distributed identically to Y. However, such a translation does not guarantee that the individual inputs and outputs x and y are paired up in a meaningful way – there are infinitely many mappings G that will induce the same distribution over $\hat{y}$. Moreover, in practice, we have found it difficult to optimize the adversarial objective in isolation: standard procedures often lead to the well-known problem of mode collapse, where all input images map to the same output image and the optimization fails to make progress [13].

These issues call for adding more structure to our objective. Therefore, we exploit the property that translation should be "cycle consistent", in the sense that if we translate, e.g., a sentence from English to French, and then translate it back from French to English, we should arrive back at the original sentence [3]. Mathematically, if we have a translator $G : X \rightarrow Y$ and another translator $F : Y \rightarrow X$, then G and F should be inverses of each other, and both mappings should be bijections. We apply this structural assumption by training both the mapping G and F simultaneously, and adding a *cycle consistency loss* [60] that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Combining this loss with adversarial losses on domains X and Y yields our full objective for unpaired image-to-image translation.

We apply our method to a wide range of applications, including style transfer, object transfiguration, attribute transfer and photo enhancement. We also compare against previous approaches that rely either on hand-defined factorizations of style and content, or on shared embedding functions, and show that our method outperforms these baselines. Our code is available at https://github.com/junyanz/CycleGAN. Check out the full version of the paper at https://arxiv.org/abs/1703.10593.

2. Related work

Generative Adversarial Networks (GANs) [14, 58] have achieved impressive results in image generation [5, 35], image editing [61], and representation learning [35, 39, 33]. Recent methods adopt the same idea for conditional image generation applications, such as text2image [36], image inpainting [34], and future prediction [32], as well as to other domains like videos [50] and 3D models [53]. The key to GANs' success is the idea of an *adversarial loss* that forces the generated images to be, in principle, indistinguishable from real images. This is particularly powerful for image generation tasks, as this is exactly the objective that much of computer graphics aims to optimize. We adopt an adversarial loss to learn the mapping such that the translated image cannot be distinguished from images in the target domain.

Image-to-Image Translation The idea of image-to-image translation goes back at least to Hertzmann et al.'s Image Analogies [17], who employ a nonparametric texture model [8] on a single input-output training image pair. More

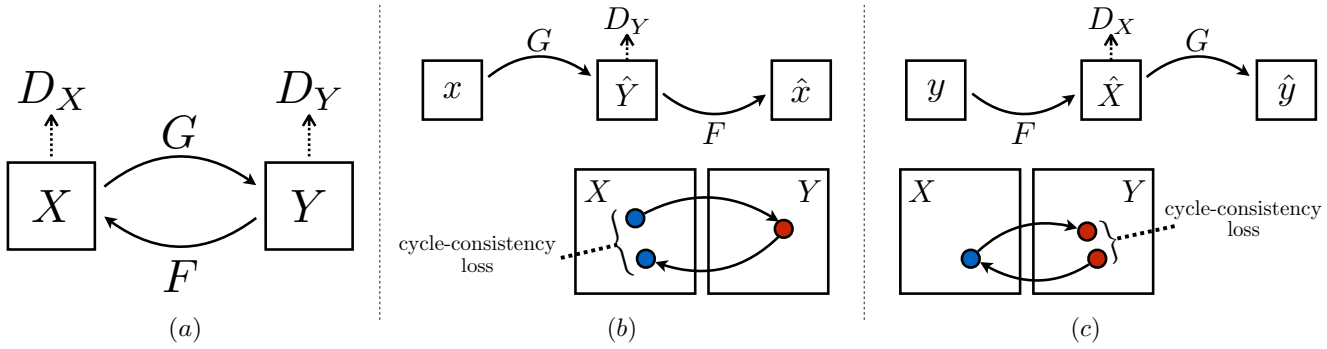

Figure 3: (a) Our model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X, F, and X. To further regularize the mappings, we introduce two "cycle consistency losses" that capture the intuition that if we translate from one domain to the other and back again we should arrive where we started: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

recent approaches use a *dataset* of input-output examples to learn a parametric translation function using CNNs, e.g. [29]. Our approach builds on the "pix2pix" framework of Isola et al. [20], which uses a conditional generative adversarial network [14] to learn a mapping from input to output images. Similar ideas have been applied to various tasks such as generating photographs from sketches [40] or from attribute and semantic layouts [22]. However, unlike these prior works, we learn the mapping without paired training examples.

Unpaired Image-to-Image Translation Several other methods also tackle the unpaired setting, where the goal is to relate two data domains, X and Y. Rosales et al. [37] propose a Bayesian framework that includes a prior based on a patch-based Markov random field computed from a source image, and a likelihood term obtained from multiple style images. More recently, CoupledGANs [28] and cross-modal scene networks [1] use a weight-sharing strategy to learn a common representation across domains. Concurrent to our method, Liu et al. [27] extends this framework with a combination of variational autoencoders [23] and generative adversarial networks. Another line of concurrent work [42, 45, 2] encourages the input and output to share certain "content" features even though they may differ in "style". They also use adversarial networks, with additional terms to enforce the output to be close to the input in a predefined metric space, such as class label space [2], image pixel space [42], and image feature space [45].

Unlike the above approaches, our formulation does not rely on any task-specific, predefined similarity function between the input and output, nor do we assume that the input and output have to lie in the same low-dimensional embedding space. This makes our method a general-purpose solution for many vision and graphics tasks. We directly compare against several prior approaches in Section 5.1. Concurrent with our work, in these same proceedings, Yi et al. [55] independently introduce a similar objective for unpaired image-to-image translation, inspired by dual learning in machine translation [15].

Cycle Consistency The idea of using transitivity as a way to regularize structured data has a long history. In visual tracking, enforcing simple forward-backward consistency has been a standard trick for decades [44]. In the language domain, verifying and improving translations via "back translation and reconsiliation" is a technique used by human translators [3] (including, humorously, by Mark Twain [47]), as well as by machines [15]. More recently, higher-order cycle consistency has been used in structure from motion [56], 3D shape matching [19], co-segmentation [51], dense semantic alignment [59, 60], and depth estimation [12]. Of these, Zhou et al. [60] and Godard et al. [12] are most similar to our work, as they use a *cycle consistency loss* as a way of using transitivity to supervise CNN training. In this work, we are introducing a similar loss to push G and F to be consistent with each other.

Neural Style Transfer [11, 21, 48, 10] is another way to perform image-to-image translation, which synthesizes a novel image by combining the content of one image with the style of another image (typically a painting) by matching the Gram matrix statistics of pre-trained deep features. Our main focus, on the other hand, is learning the mapping between two domains, rather than between two specific images, by trying to capture correspondences between higher-level appearance structures. Therefore, our method can be applied to other tasks, such as painting→ photo, object transfiguration, etc. where single sample transfer methods do not perform well. We compare these two methods in Section 5.2.

3. Formulation

Our goal is to learn mapping functions between two domains X and Y given training samples $\{x_i\}_{i=1}^N \in X$ and $\{y_j\}_{j=1}^M \in Y$. As illustrated in Figure 3 (a), our model includes two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$. In addition, we introduce two adversarial discriminators D_X and D_Y, where D_X aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$; in the same way, D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$. Our objective contains kinds of two terms: *adversarial losses* [14] for matching the distribution of generated images to the data distribution in

the target domain; and a *cycle consistency loss* to prevent the learned mappings G and F from contradicting each other.

3.1. Adversarial Loss

We apply adversarial losses [14] to both mapping functions. For the mapping function $G : X \rightarrow Y$ and its discriminator D_Y, we express the objective as:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) =& \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],\end{aligned} \tag{1}$$

where G tries to generate images $G(x)$ that look similar to images from domain Y, while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. We introduce a similar adversarial loss for the mapping function $F : Y \rightarrow X$ and its discriminator D_X as well: i.e. $\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$.

3.2. Cycle Consistency Loss

Adversarial training can, in theory, learn mappings G and F that produce outputs identically distributed as target domains Y and X respectively (strictly speaking, this requires G and F to be stochastic functions) [13]. However, with large enough capacity, a network can map the same set of input images to any random permutation of images in the target domain, where any of the learned mappings can induce an output distribution that matches the target distribution. To further reduce the space of possible mapping functions, we argue that the learned mapping functions should be cycle-consistent: as shown in Figure 3 (b), for each image x from domain X, the image translation cycle should be able to bring x back to the original image, i.e. $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. We call this *forward cycle consistency*. Similarly, as illustrated in Figure 3 (c), for each image y from domain Y, G and F should also satisfy *backward cycle consistency*: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. We can incentivize this behavior using a *cycle consistency loss*:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) =& \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned} \tag{2}$$

In preliminary experiments, we also tried replacing the L1 norm in this loss with an adversarial loss between $F(G(x))$ and x, and between $G(F(y))$ and y, but did not observe improved performance. The behavior induced by the cycle consistency loss can be observed in the arXiv version.

3.3. Full Objective

Our full objective is:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) =& \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned} \tag{3}$$

where λ controls the relative importance of the two objectives. We aim to solve:

$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y). \tag{4}$$

Notice that our model can be viewed as training two "autoencoders" [18]: we learn one autoencoder $F \circ G : X \rightarrow X$ jointly with another $G \circ F : Y \rightarrow Y$. However, these autoencoders each have special internal structure: they map an image to itself via an intermediate representation that is a translation of the image into another domain. Such a setup can also be seen as a special case of "adversarial autoencoders" [30], which use an adversarial loss to train the bottleneck layer of an autoencoder to match an arbitrary target distribution. In our case, the target distribution for the $X \rightarrow X$ autoencoder is that of domain Y. In Section 5.1.3, we compare our method against ablations of the full objective, and empirically show that both objectives play critical roles in arriving at high-quality results.

4. Implementation

Network Architecture We adapt the architecture for our generative networks from Johnson et al. [21] who have shown impressive results for neural style transfer and super-resolution. This network contains two stride-2 convolutions, several residual blocks [16], and two $\frac{1}{2}$-strided convolutions. Similar to Johnson et al. [21], we use instance normalization [49]. For the discriminator networks we use 70×70 PatchGANs [20, 26, 25], which aim to classify whether 70×70 overlapping image patches are real or fake. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator, and can be applied to arbitrarily-sized images in a fully convolutional fashion [20].

Training details We apply two techniques from recent works to stabilize our model training procedure. First, for $\mathcal{L}_{GAN}$ (Equation 1), we replace the negative log likelihood objective by a least square loss [31]. This loss performs more stably during training and generates higher quality results. Equation 1 then becomes:

$$\begin{aligned}\mathcal{L}_{\text{LSGAN}}(G, D_Y, X, Y) =& \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_Y(y) - 1)^2] \\ &+ \mathbb{E}_{x \sim p_{\text{data}}(x)}[D_Y(G(x))^2],\end{aligned} \tag{5}$$

Second, to reduce model oscillation [13], we follow Shrivastava et al's strategy [42] and update the discriminators D_X and D_Y using a history of generated images rather than the ones produced by the latest generative networks. We keep an image buffer that stores the 50 previously generated images.

Please refer to our arXiv paper for more details about the datasets, architectures and training procedures.

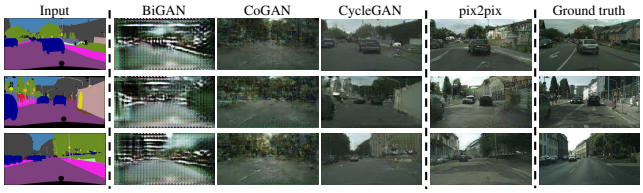

Figure 4: Different methods for mapping labels→photos trained on cityscapes. From left to right: input, BiGAN/ALI [6, 7], CoGAN [28], CycleGAN (ours), pix2pix [20] trained on paired data, and ground truth.

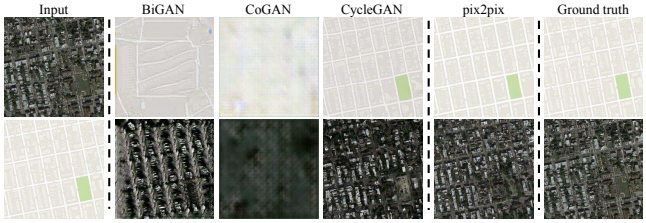

Figure 5: Different methods for mapping aerial photos↔maps on Google Maps. From left to right: input, BiGAN/ALI [6, 7], CoGAN [28], CycleGAN (ours), pix2pix [20] trained on paired data, and ground truth.

5. Results

We first compare our approach against recent methods for unpaired image-to-image translation on paired datasets where ground truth input-output pairs are available for evaluation. We then study the importance of both the adversarial loss and the cycle consistency loss, and compare our full method against several variants. Finally, we demonstrate the generality of our algorithm on a wide range of applications where paired data does not exist. For brevity, we refer to our method as `CycleGAN`.

5.1. Evaluation

Using the same evaluation datasets and metrics as "pix2pix" [20], we compare our method against several baselines both qualitatively and quantitatively. We also perform ablation study on the full loss function.

5.1.1 Baselines

CoGAN [28] This method learns one GAN generator for domain X and one for domain Y, with shared weights on the first few layers for shared latent representation. Translation from X to Y can be achieved by finding a latent representation that generates image X and then rendering this latent representation into style Y.

Pixel loss + GAN [42] Like our method, Shrivastava et al. [42] uses an adversarial loss to train a translation from X to Y. The regularization term $\|X - \hat{Y}\|_1$ was used to penalize making large changes at pixel level.

Feature loss + GAN We also test a variant of [42] where the L1 loss is computed over deep image features using a pretrained network (VGG-16 `relu4_2` [43]), rather than over RGB pixel values.

BiGAN/ALI [7, 6] Unconditional GANs [14] learn a generator $G : Z \rightarrow X$, that maps random noise Z to images X. The BiGAN [7] and ALI [6] propose to also learn the inverse mapping function $F : X \rightarrow Z$. Though they were originally designed for mapping a latent vector z to an image x, we implemented the same objective for mapping a source image x to a target image y.

pix2pix [20] We also compare against pix2pix [20], which is trained on paired data, to see how close we can get to this "upper bound" without using paired data.

For fair comparison, we implement all the baselines using the same architecture and details as our method except for CoGAN [28]. We use the public implementation of CoGAN due to fundamental differences in architecture [1].

Loss	Map → Photo % Turkers labeled *real*	Photo → Map % Turkers labeled *real*
CoGAN [28]	0.6% ± 0.5%	0.9% ± 0.5%
BiGAN/ALI [7, 6]	2.1% ± 1.0%	1.9% ± 0.9%
Pixel loss + GAN [42]	0.7% ± 0.5%	2.6% ± 1.1%
Feature loss + GAN	1.2% ± 0.6%	0.3% ± 0.2%
CycleGAN (ours)	**26.8%** ± **2.8%**	**23.2%** ± **3.4%**

Table 1: AMT "real vs fake" test on maps↔aerial photos.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [28]	0.40	0.10	0.06
BiGAN/ALI [7, 6]	0.19	0.06	0.02
Pixel loss + GAN [42]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [20]	0.71	0.25	0.18

Table 2: FCN-scores for different methods, evaluated on Cityscapes labels→photos.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [28]	0.45	0.11	0.08
BiGAN/ALI [7, 6]	0.41	0.13	0.07
Pixel loss + GAN [42]	0.47	0.11	0.07
Feature loss + GAN	0.50	0.10	0.06
CycleGAN (ours)	**0.58**	**0.22**	**0.16**
pix2pix [20]	0.85	0.40	0.32

Table 3: Classification performance of photo→labels for different methods on cityscapes.

5.1.2 Comparison against baselines

As can be seen in Figure 4 and Figure 5, we were unable to achieve compelling results with any of the baselines. Our method, on the other hand, is able to produce translations that are often of similar quality to the fully supervised pix2pix. We exclude pixel loss + GAN and feature loss + GAN in the figures, as both of the methods fail to produce results at all close to the target domain (full results can be viewed at `https://junyanz.github.io/CycleGAN/`).

In addition, our method and the baselines are quantitatively compared in three ways. First, we run "real vs fake" study on Amazon Mechanical Turk (AMT) workers to assess perceptual realism [20]. Second, we train photo→label task on the

[1] https://github.com/mingyuliutw/CoGAN

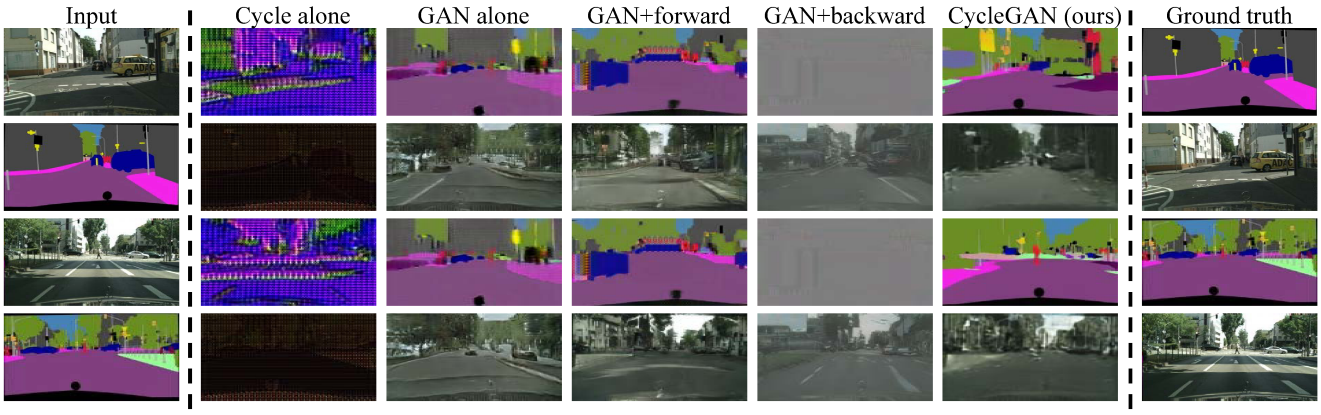

Figure 6: Different variants of our method for mapping labels↔photos trained on cityscapes. From left to right: input, cycle-consistency loss alone, adversarial loss alone, GAN + forward cycle-consistency loss ($F(G(x)) \approx x$), GAN + backward cycle-consistency loss ($G(F(y)) \approx y$), CycleGAN (our full method), and ground truth. Both *Cycle alone* and *GAN + backward* fail to produce images similar to the target domain. *GAN alone* and *GAN + forward* suffer from mode collapse, producing identical label maps regardless of the input photo.

Cityscapes dataset, and compare the output label images with the ground truth using the standard metrics on the Cityscapes benchmark [4]. Lastly, we train label→photo task on the same dataset and evaluate the output photos using an off-the-shelf fully-convolutional semantic segmentation network [29]. We find that our method significantly outperforms the baselines in all three experiments. Table 1 reports performance on the AMT perceptual realism task. Here, we see that our method can fool participants on around a quarter of trials, in both the map→photo direction and the photo→map direction. All baselines almost never fooled participants. Table 2 and Table 3 assess the performance of the label↔photo task on the Cityscapes. In both cases, our method again outperforms the baselines. Detailed procedures and results of each experiment can be found in our arXiv version.

5.1.3 Ablation Study

We compare against ablations of our full loss. Figure 6 shows several qualitative examples. Removing the GAN loss substantially degrades results, as does removing the cycle-consistency loss. We therefore conclude that both terms are critical to our results. We also evaluate our method with the cycle loss in only one direction: GAN+forward cycle loss $\mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1]$, or GAN+backward cycle loss $\mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$ (Equation 2) and find that it often incurs training instability and causes mode collapse, especially for the direction of the mapping that was removed. We also quantitatively measured the ablations on Cityscapes photos→label, whose results can be found in our arXiv version.

5.2. Applications

We demonstrate our method on several applications where paired training data does not exist.We observe that translations on training data are often more appealing than those on test data, and full results of all applications on both training and test data can be viewed on our project website.

Object transfiguration (Figure 7) The model is trained to translate one object class from Imagenet [38] to another (each class contains around 1000 training images). Turmukhambetov et al.[46] proposes a subspace model to translate one object into another object of the same category, while our method focuses on object transfiguration between two visually similar categories.

Season transfer (Figure 7) The model is trained on the winter and summer photos of Yosemite on Flickr.

Collection style transfer (Figure 8) We train the model on landscape photographs downloaded from Flickr and WikiArt. Note that unlike recent work on "neural style transfer" [11], our method learns to mimic the style of an entire *set* of artworks (e.g. Van Gogh), rather than transferring the style of a single selected piece of art (e.g. Starry Night). In Figure 5.2, we compare our results with [11].

Photo generation from paintings (Figure 9) For painting→photo, we find that it is helpful to introduce an additional loss to encourage the mapping to preserve color composition between the input and output. In particular, we adopt the technique of Taigman et al. [45] and regularize the generator to be near an identity mapping when real samples of the target domain are provided as the input to the generator: i.e. $\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(y) - y\|_1] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(x) - x\|_1]$.

Without $\mathcal{L}_{\text{identity}}$, the generator G and F are free to change the tint of input images when there is no need to. For example, when learning the mapping between Monet's paintings and Flickr photographs, the generator often maps paintings of daytime to photographs taken during sunset, because such a mapping may be equally valid under the adversarial loss and cycle consistency loss. The effect of this *identity mapping loss* can be found in our arXiv paper.

In Figure 9, we show additional results translating Monet

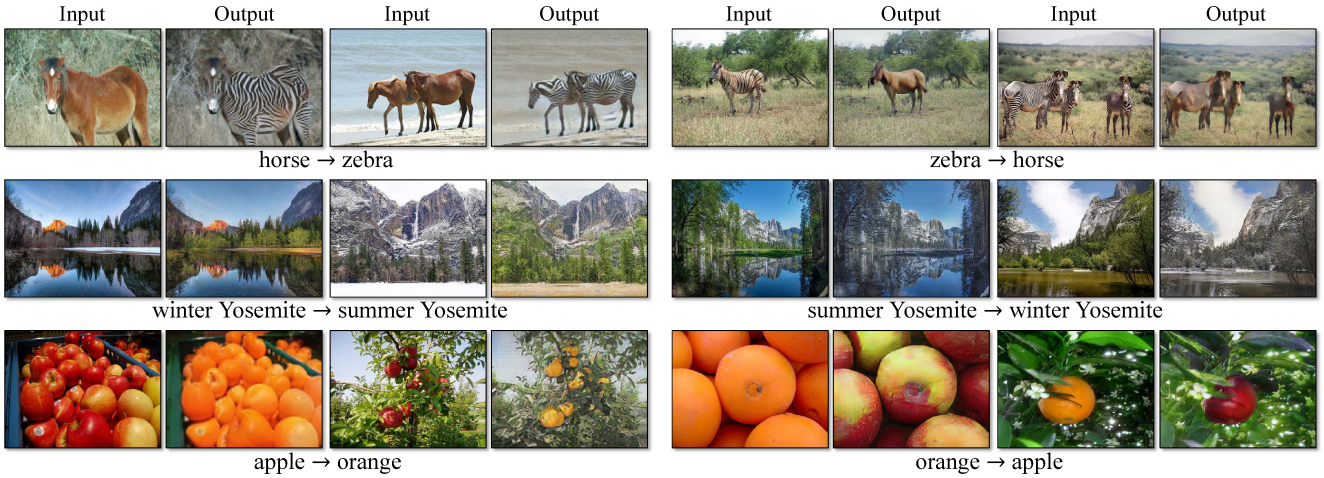

Figure 7: Results on several translation problems. These images are relatively successful results – please see our website for more comprehensive results.

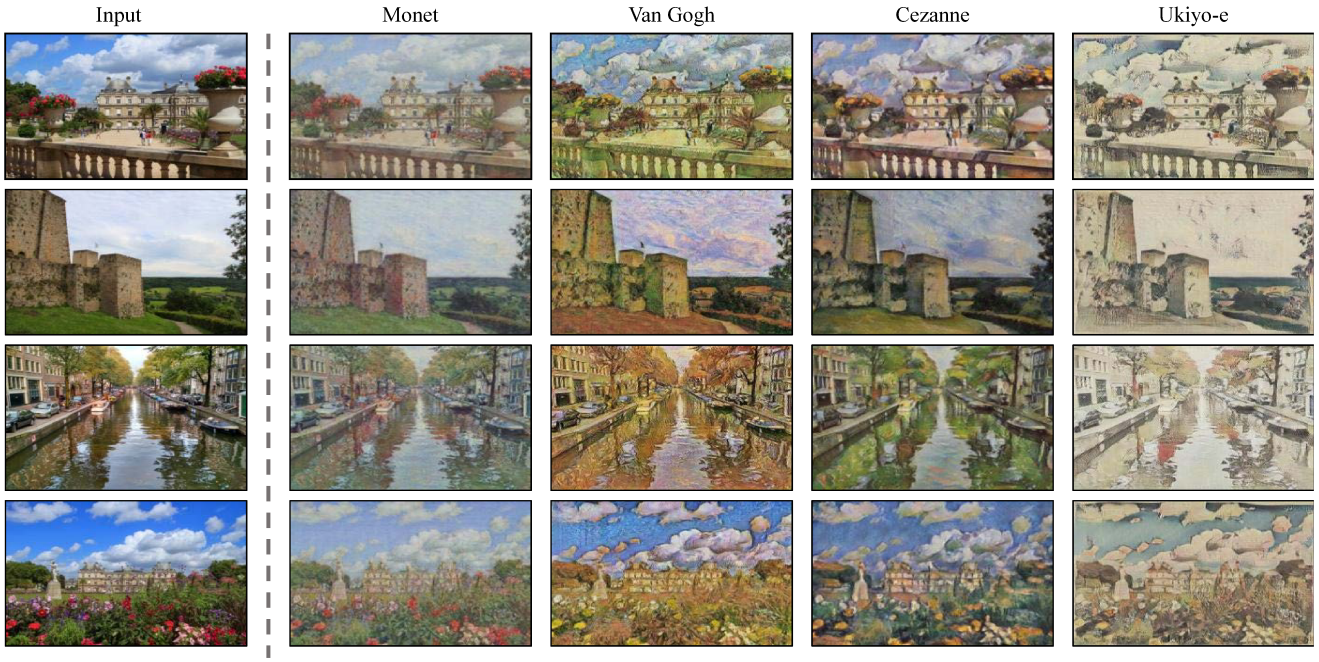

Figure 8: We transfer input images into different artistic styles. Please see our website for additional examples.

paintings to photographs. This figure shows results on paintings that were included in the *training set*, whereas for all other experiments in the paper, we only evaluate and show test set results. Because the training set does not include paired data, coming up with a plausible translation for a training set painting is a nontrivial task. Indeed, since Monet is no longer able to create new paintings, generalization to unseen, "test set", paintings is not a pressing problem.

Photo enhancement (Figure 7) We show that our method can be used to generate photos with shallower depth of field. We train the model on flower photos downloaded from Flickr. The source domain consists of photos of flower taken by smartphones, which usually have deep depth of field due to a small aperture. The target photos were taken with DSLRs with a larger aperture. Our model successfully generates photos with shallower depth of field from the photos taken by smartphones.

6. Limitations and Discussion

Although our method can achieve compelling results in many cases, the results are far from uniformly positive. Several typical failure cases are shown in Figure 12. On translation tasks that involve color and texture changes, like many of those reported above, the method often succeeds. We have also explored tasks that require geometric changes, with little success. For example, on the task of dog→cat transfiguration, the learned translation degenerates to making minimal changes to the input (Figure 12). Handling more varied and extreme transformations, especially geometric changes, is an

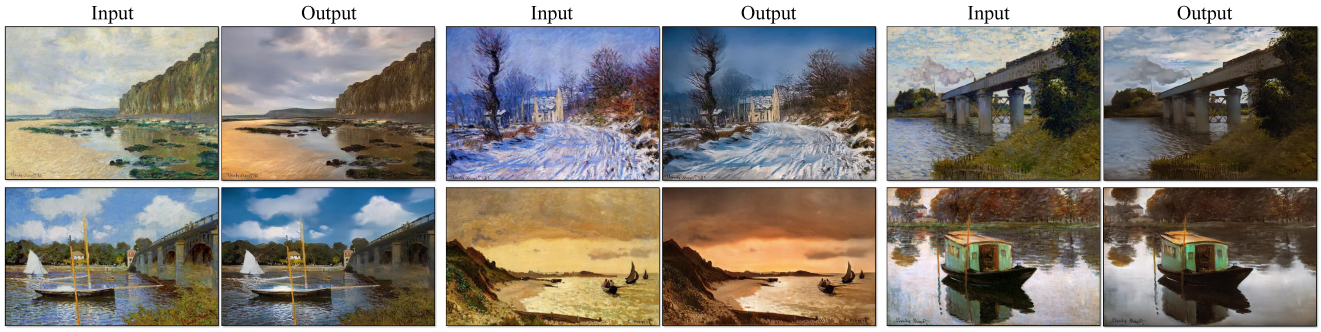

Figure 9: Results on mapping Monet paintings to photographs. Please see our website for additional examples.

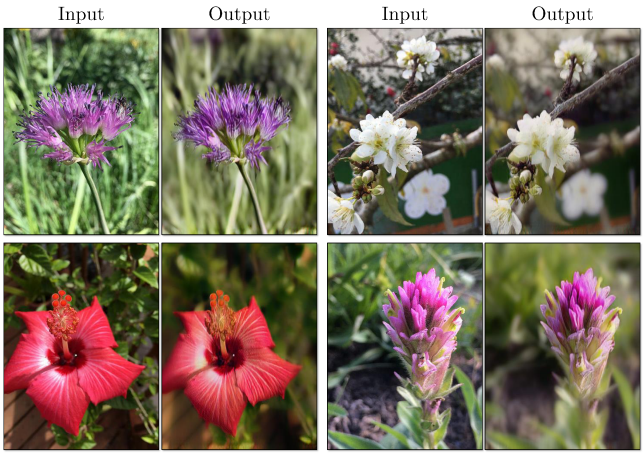

Figure 10: Photo enhancement: mapping from a set of iPhone snaps to professional DSLR photographs, the system often learns to produce shallow focus. Here we show some of the most successful results in our test set – average performance is considerably worse. Please see our website for more comprehensive and random examples.

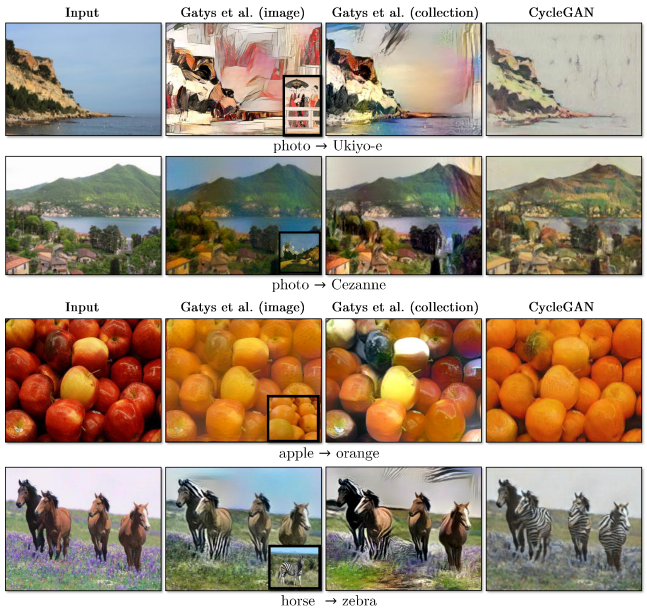

Figure 11: We compare our method with neural style transfer [11]. Left to right: input images, results from [11] using single representative image as a style image, results from [11] using all the images from the target domain, and CycleGAN (ours)

Figure 12: Some failure cases of our method.

important problem for future work.

Some failure cases are caused by the distribution characteristic of the training datasets. For example, the horse → zebra task of Figure 12 has completely failed, because our model was trained on the *wild horse, zebra* synsets of ImageNet, which does not contain images of a person riding horse or zebra.

We also observe a lingering gap between the results achievable with paired training data and those achieved by our unpaired method. In some cases, this gap may be very hard – or even impossible – to close: for example, our method sometimes permutes the labels for tree and building in the output of the photos→labels task. To resolve this ambiguity may require some form of weak semantic supervision. Integrating weak or semi-supervised data may lead to substantially more powerful translators, still at a fraction of the annotation cost of the fully-supervised systems.

Nonetheless, in many cases completely unpaired data is plentifully available and should be made use of. This paper pushes the boundaries of what is possible in this "unsupervised" setting.

Acknowledgments We thank Aaron Hertzmann, Shiry Ginosar, Deepak Pathak, Bryan Russell, Eli Shechtman, Richard Zhang, and Tinghui Zhou for many helpful comments. This work was supported in part by NSF SMA-1514512, NSF IIS-1633310, a Google Research Award, Intel Corp, and hardware donations from NVIDIA. JYZ is supported by the Facebook Graduate Fellowship and TP is supported by the Samsung Scholarship. The photographs used in style transfer were taken by AE, mostly in France.

References

[1] Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash, and A. Torralba. Cross-modal scene networks. *arXiv preprint arXiv:1610.09003*, 2016. 3

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016. 3

[3] R. W. Brislin. Back-translation for cross-cultural research. *Journal of cross-cultural psychology*, 1(3):185–216, 1970. 2, 3

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 6

[5] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015. 2

[6] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 5

[7] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 5

[8] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, volume 2, pages 1033–1038. IEEE, 1999. 2

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015. 2

[10] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*, 2016. 3

[11] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. *CVPR*, 2016. 3, 6, 8

[12] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 3

[13] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 2, 4

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2, 3, 4, 5

[15] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W.-Y. Ma. Dual learning for machine translation. In *NIPS*, pages 820–828, 2016. 3

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[17] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *SIGGRAPH*, pages 327–340. ACM, 2001. 2

[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 4

[19] Q.-X. Huang and L. Guibas. Consistent shape maps via semidefinite programming. In *Computer Graphics Forum*, volume 32, pages 177–186. Wiley Online Library, 2013. 3

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3, 4, 5

[21] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 2, 3, 4

[22] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016. 3

[23] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2014. 3

[24] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)*, 33(4):149, 2014. 2

[25] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016. 4

[26] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV*, 2016. 4

[27] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 3

[28] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, pages 469–477, 2016. 3, 5

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2, 3, 6

[30] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 4

[31] 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. 4

[32] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016. 2

[33] M. F. Mathieu, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *NIPS*, pages 5040–5048, 2016. 2

[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016. 2

[35] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[36] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[37] R. Rosales, K. Achan, and B. J. Frey. Unsupervised image translation. In *iccv*, pages 472–478, 2003. 3

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016. 2

[40] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017. 3

[41] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 32(6):200, 2013. 2

[42] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016. 3, 4, 5

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[44] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, pages 438–451. Springer, 2010. 3

[45] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 3, 6

[46] D. Turmukhambetov, N. D. Campbell, S. J. Prince, and J. Kautz. Modeling object appearance using context-conditioned component analysis. In *CVPR*, pages 4156–4164, 2015. 6

[47] M. Twain. *The Jumping Frog: in English, then in French, and then Clawed Back into a Civilized Language Once More by Patient, Unremunerated Toil*. 1903. 3

[48] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Int. Conf. on Machine Learning (ICML)*, 2016. 3

[49] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[50] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, pages 613–621, 2016. 2

[51] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, pages 849–856, 2013. 3

[52] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. *ECCV*, 2016. 2

[53] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, pages 82–90, 2016. 2

[54] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2

[55] Z. Yi, H. Zhang, T. Gong, Tan, and M. Gong. Dual-gan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. 3

[56] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433. IEEE, 2010. 3

[57] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016. 2

[58] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[59] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, pages 1191–1200, 2015. 3

[60] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, pages 117–126, 2016. 2, 3

[61] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 2