ESTHER[1]: Extremely Simple Image Translation Through Self-Regularization

Chao Yang[1]
harryyang.hk@gmail.com
Taehwan Kim[2]
taehwan@oben.com
Ruizhe Wang[2]
ruizhe@oben.com
Hao Peng[2]
hpeng@oben.com
C.-C. Jay Kuo[1]
cckuo@sipi.usc.edu

[1] University of Southern California, Los Angeles, USA
[2] ObEN, Inc. 130 W Union St, Pasadena, Los Angeles, USA

Abstract

Image translation between two domains is a class of problems where the goal is to learn the mapping from an input image in the source domain to an output image in the target domain. It has important applications such as data augmentation, domain adaptation, and unsupervised training. When paired training data are not accessible, the mapping between the two domains is highly under-constrained and we are faced with an ill-posed task. Existing approaches tackling this challenge usually make assumptions and introduce prior constraints. For example, CycleGAN [59] assumes cycle-consistency while UNIT [31] assumes shared latent-space between the two domains. We argue that none of these assumptions explicitly guarantee that the learned mapping is the desired one. We, taking a step back, observe that most image translations are based on the intuitive requirement that the translated image needs to be perceptually similar to the original image and also appear to come from the new domain. On the basis of such observation, we propose an extremely simple yet effective image translation approach, which consists of a single generator and is trained with a self-regularization term and an adversarial term. We further propose an adaptive method to search for the best weight between the two terms. Extensive experiments and evaluations show that our model is significantly more cost-effective and can be trained under budget, yet easily achieves better performance than other methods on a broad range of tasks and applications.

1 Introduction

Many computer vision problems can be cast as an image-to-image translation problem, whose task is to map an image in one domain to a corresponding image in another domain. For example, image colorization can be considered as mapping a gray-scale image to a corresponding image in RGB space [57]; style transfer can be viewed as translating an image in one style to a corresponding image with another style [12, 13, 20]. Other tasks

[1] Esther is an English name and means "star" in Persian.

falling into this category include semantic segmentation [33], super-resolution [27], image manipulation [19], etc. Another important application of image translation is related to domain adaptation and unsupervised learning: with the rise of deep learning, it is now considered crucial to have large labeled training datasets such as ImageNet [43] and COCO [28]. However, labeling and annotating such large datasets are expensive and not scalable. An alternative is to use synthetic or simulated data for training, whose labels are trivial to acquire [2, 21, 35, 39, 41, 44, 51, 60]. Unfortunately, learning from synthetic data can be problematic and most of the time does not generalize to real-world data, due to the statistical gap between the two domains. Furthermore, for deep neural networks which are powerful to learn small details, it is anticipated that the trained model easily over-fits to the synthetic domain. In order to close this gap, we can either find mappings or domain-invariant representations at feature level [1, 4, 6, 11, 15, 22, 34, 49, 52] or learn to translate images from one domain to another domain to create "fake" labeled data for training [5, 27, 30, 31, 56, 59]. In the latter case, we usually hope to learn a mapping that preserves the labels and as well as the attributes we care about most.

Typically there exist two settings for image translation given two domains X and Y. The first setting is supervised, where example image pairs x, y are available. This means for the training data, for each image $x_i \in X$ there is a corresponding $y_i \in Y$, and we wish to find a translator $G : X \to Y$ such that $G(x_i) \approx y_i$. Representative translation systems in the supervised setting include domain-specific works [10, 18, 25, 33, 46, 54, 55, 57] and the more general Pix2Pix [19, 53]. However, paired training data is rarely easy to acquire. For example, for image stylization, obtaining paired data requires lengthy artist authoring and is extremely expensive. For other tasks like object transfiguration, the desired output is not even well defined.

Therefore, we focus on the second setting, which is unsupervised image translation. In the unsupervised setting, X and Y are two independent sets of images, and we do not have access of paired examples showing how an image $x_i \in X$ could be translated to an image $y_i \in Y$. Our task is then to seek an algorithm that can learn to translate between X and Y without desired input-output examples. The unsupervised image translation setting potentially has greater applications because of its simplicity and flexibility to use but is also much more difficult. In fact, it is a highly under-constrained and ill-posed problem, since there could be unlimited number of mappings between X and Y: from the probabilistic view, the challenge is to learn a joint distribution of images in different domains. As stated by the coupling theory [29], there exists an infinite set of joint distributions that can arrive the two marginal distributions in two different domains. Therefore, additional assumptions and constraints are needed for us to exploit the structure and supervision necessary to learn the mapping.

Existing works that address this problem assume that there is some relationship between the two domains. For example, CycleGAN [59] assumes cycle-consistency and the existence of an inverse mapping F that translates from Y to X. It then trains two generators which are bijections and inverse to each other and uses adversarial constraint [14] to ensure the translated images appear to be drawn from the target domain and the cycle-consistency constraint to ensure the translated image can be mapped back to the original image using the inverse mapping ($F(G(x)) \approx x$ and $G(F(y)) \approx y$). UNIT [31], on the other hand, assumes shared-latent space, meaning a pair of images in different domains can be mapped to some shared latent representations. The model trains two generators G_X, G_Y with shared layers. Both G_X and G_Y map an input to itself, while the domain translation is realized by letting x_i go through part of G_X and part of G_Y to get y_i. The model is trained with an adversarial constraint on the image, a variational constraint on the latent code [24, 40], and another cycle-consistency constraint.

Assuming cycle-consistency clearly ensures 1-1 mapping and avoids mode collapses [45], and both models generate reasonable image translation and domain adaptation results. However, there are several issues with such approaches. First and foremost, cycle-consistency does not guarantee that the mapping learned is the desired mapping. Theoretically, CycleGAN could find any arbitrary 1-1 mapping that satisfies the constraints. Having multiple global optima is problematic since, in our experiments, we observed that the training is far from stable, meaning it does not guarantee to converge or reproduce the same results every time. Second, there is a sensitive trade-off between the faithfulness of the translated image to the input image and how similar it resembles the new domain, and it requires excessive manual tuning of the weight between the adversarial loss and the reconstruction loss to get satisfying results. Third, most of the time we only care about one-way translation, while CycleGAN always requires the training of two generators that are bijections. This is not only cumbersome but it is also hard to balance the effects of the two generators.

We propose a much simpler yet more effective image translation model called **E**xtremely **S**imple image **T**ranslation t**H**rough s**E**lf-**R**egularization (**ESTHER**). We first re-consider what should be the desired output of an image translation task: most of the time the desired output should not only resemble the target domain but also preserve certain attributes and share similar visual appearance with input. For example, in the case of horse-zebra translation [59], the output zebra should be similar to the input horse in terms of the scene background, the location and the shape of the zebra/horse, and the only thing that we wish to add is the black-white stripes that specifically belong to a zebra. In the domain adaptation task that translates MNIST [26] to USPS [9], we expect the output is visually similar to the input in terms of the shape and structure of the digit such that it preserves the label. Based on such observation, our model proposes to use a single generator that maps X to Y and is trained with a self-regularization term that enforces perceptual similarity between the output and the input, together with an adversarial term that enforces the output to appear like drawn from Y. We further propose an automatic and principle way to find the optimal weight between the self-regularization term and the adversarial term such that we do not have to manually search for the best hyper-parameter. Lastly, in order to create sharp and more realistic looking results, we introduce several design choices of the network architecture, which prove to improve the quality and resolution of the output images.

Our model does not rely on cycle-consistency or shared representation assumption, and it only learns one-way mapping. This makes training of the model simpler and more efficient. Although the constraint is susceptible to oversimplify certain scenarios, we found that the model works surprisingly well in a wide variety of image translation and domain adaptation tasks, achieving superior qualitative and quantitative results. In the end, we discuss the face 3D morphable model [2] prediction as a useful real-world application of our approach. We also carefully analyze the relation between our approach and CycleGAN/UNIT and argue that neither cycle-consistency or shared representation is a necessary assumption.

2 Our Method

We begin by explaining our model for unsupervised image translation. Let X and Y be two image domains, our goal is to train a generator $G_\theta : X \to Y$, where θ are the function parameters. We are given unpaired samples x and y, and the unsupervised setting assumes that x and y are independently drawn from the marginal distributions $P_{x\sim X}(x)$ and $P_{y\sim Y}(y)$. Let $y' = G_\theta(x)$ denote the translated image, the key requirement is that y' should appear like drawn from domain Y, while preserving the low-level visual characteristics of x. The

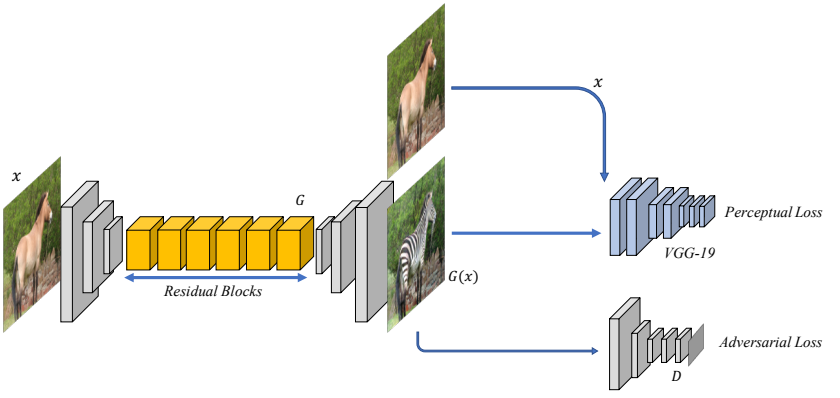


Figure 1: Model overview. Our model consists of a single generator G that maps an image from one domain to another domain. We train G using the self-regularization perceptual loss and the adversarial loss.

translated images y' can be further used for other downstream tasks such as unsupervised learning but in our case, we decouple image translation from its applications.

Based on the requirements described, we propose to learn θ by minimizing the following losses:

$$\mathcal{L}_G(\theta) = \ell_{adv}(G_\theta(x), Y) + \lambda \ell_{reg}(x, G_\theta(x)). \tag{1}$$

The first part of the losses, ℓ_{adv}, is the adversarial loss on the image domain that makes sure that $G_\theta(x)$ appears like domain Y. The second part of the losses ℓ_{reg} makes sure that $G_\theta(x)$ is visually similar to x. ℓ_{adv} is given by a discriminator D trained jointly with G, and ℓ_{reg} is measured with perceptual loss. Note that similar formulations have been used in [47] for simulated image refinement and [20] for style transfer, however no existing works have generalized it as a principal framework for general image translation and domain adaptation tasks, neglecting its unique power that can solve a much wider range of problems.

The model architectures: Our model consists of a generator G and a discriminator D. The generator G is based on Fully Convolutional Network (FCN) and leverages properties of convolutional neural networks, such as translation invariance and parameter sharing. Similar to [19, 59], the generator G is built with three components: a down-sampling front-end to reduce the size, followed by multiple residual blocks [17], and an up-sampling back-end to restore the original dimensions. The down-samping front-end consists of three convolutional layers, each with a stride of 2. The intermediate part contains nine residual blocks that keep the height/width constant, and the up-sampling back-end consists of three transposed convolution, also with a stride of 2. Each convolutional layer is followed by batch normalization and ReLU activation, except for the last layer whose output is in the image space. Using down-sampling at the beginning increases the receptive field of the residual blocks and makes it easier to learn the transformation at a smaller scale. Another modification is that we adopt the dilated convolution in all residual blocks, and set the dilation factor to 2. Dilated convolutions use spaced kernels, enabling it to compute each output value with a wider view of input without increasing the number of parameters and computational burden. For the discriminator, we use a five-layer convolutional network. The first three layers have a stride of 2 followed by two convolution layers with stride 1, which effectively down-samples

the networks three times. The output is a vector of real/fake predictions and each value corresponds to a patch of the image. Classifying each patch as real/fake introduces PatchGAN, and is shown to work better than the global GAN [19, 59].

The image domain adversarial loss: The vanilla Generative Adversarial Network [14] loss plays a two-player min-max game to update the network G and D. G learns to translate the image x to y' which appears as if it is from Y, while D learns to distinguish y, which is a real image from Y, from the generated image y'. The parameters of D and G are updated alternatively. The discriminator D updates its parameters by minimizing the following loss:

$$\mathcal{L}_D(\phi) = \log(D_\phi(y')) - \log(\overrightarrow{1} - D_\phi(y)). \tag{2}$$

Here ϕ is the parameters of D. Note that here D outputs a vector of predictions instead of one values. The image domain adversarial loss used to update the generator G is defined as:

$$\ell_{adv}(G_\theta(x), Y) = -\log(\overrightarrow{1} - D_\phi(G_\theta(x))). \tag{3}$$

By minimizing the loss function, the generator G learns to create translated image that fools the network D into classifying the image as sampled from Y. Jointly training G and D with constraints (2), (3) yields to minimize the following combined GAN objective:

$$\mathcal{L}_{GAN}(G,D,X,Y) = E_{y\sim Y}[\log D(y)] + E_{x\sim X}[\log(1 - D(G(x)))]. \tag{4}$$

The self-regularization loss: Theoretically, adversarial training can learn a mapping G that produces outputs identically distributed as the target domain Y. However, if the capacity is large enough, a network can map the input images to any random permutations of images in the target domain. Thus, adversarial loses alone cannot guarantee that the learned function G maps the input x to the desired output y. To further constrain the learning mapping such that it is meaningful, we argue that the mapping function G should preserve visual characteristics of the input image. In other words, the output and the input need to share perceptual similarities, especially regarding the low-level features. Such features may include color, edges, shape, objects, etc. We impose this constraint with the self-regularization term, which is modeled by minimizing the distance between the translated image y' and the input x: $\ell_{reg} = d(x, G(x))$. Here d is some distance function d, which can be ℓ_2, ℓ_1, SSIM, etc. However, recent research suggests that using perceptual distance based on a pre-trained network corresponds much better to human perception of similarity comparing with traditional distance measures [58]. Known as the perceptual loss, it is defined as:

$$\ell_{reg}(y', x) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} (\| w_l^T \circ (\hat{F}(x)^l_{hw} - \hat{F}(y')^l_{hw}) \|_2^2). \tag{5}$$

Here $\hat{F}$ is a pre-trained network to extract the neural features. We use l to represent each layer, and H_l, W_l are the height and width of feature $\hat{F}^l$. We compute the ℓ_2 difference at each location h, w of $\hat{F}^l$ and average over the feature height and width. In our experiment, we use VGG-19 pre-trained on ImageNet [48]. We extract neural features with $\hat{F}$ across multiple layers, each scaled by a layer-wise weight. We did extensive experiments to try different combinations of feature layers and obtained the best results by only using the first three layers of VGG-19 and set the layer-wise weight w_1, w_2, w_3 to be $1.0/32, 1.0/16, 1.0/8$ respectively. This conforms to the intuition that we would like to preserve the low-level traits of the input during translation. Note that this may not always be true (such as in texture

transfer), but it is a hyper-parameter that could be easily adjusted based on different problem settings. We also experimented with using different pre-trained networks such as AlexNet to extract neural features as suggested by [58] but do not observe much difference in results.

Adaptive weight induction: Like other image translation methods, the resemblance to the new domain and faithfulness to the original image is a trade-off. In CycleGAN and UNIT, it is controlled by the weight between the cycle-consistency constraint and the adversarial constraint. To our approach, it is determined by the weight λ of the self-regularization term relative to the image adversarial term. If λ is too large, the translated image will be close to the input but does not look like the new domain. If λ is too small, the translated image would fail to pertain the visual traits of the input. Previous approaches usually decide the weight heuristically. Here we propose an adaptive scheme to search for the best λ: we start by setting $\lambda = 0$, which means we only use the adversarial constraint to train the generator. Then we gradually increase λ. This would lead to the increase of the adversarial loss as the output would shift away from Y to X, which makes it easier for D to classify. We stop increasing λ when the adversarial loss sinks below some threshold ℓ^t_{adv}. We then keep λ constant and continue to train the network until converging. Using the adaptive weight induction scheme avoids manual tuning of λ for each specific task and gives results that are *both similar to the input x and the new domain Y*.

Even though our approach assumes the perceptual distance between x and its corresponding $y \in Y$ is small, our approach generalizes well to tasks where the input and output domains are significantly different, such as translation of photo to map, day to night, etc., as long as our assumption generally holds. For example, in the case of photo to map, the park (photo) is labeled as green (map) and the water (photo) is labeled as blue (map), which provides certain low-level similarities. We additionally observe that our results are consistently similar or better than CycleGAN/UNIT, even in the extreme case when we swap the color of the map labels to deliberately destroy the perceptual similarity, showing that using a more general cycle-consistency assumption does not benefit the performance of image translation. Note that our approach is a meta-algorithm, and we could potentially improve the results by using new/more advanced components. For example, the generator and discriminator could be easily replaced with the latest GAN architectures such as LSGAN [36], WGAN-GP [16], or adding spectral normalization [37]. We may also improve the results by employing a more specific self-regularizaton term that is fine-tuned on the datasets we work on.

3 Results

We tested our model on a variety of datasets and tasks. In the following, we show the qualitative results of image translation, as well as quantitative results in several domain adaptation settings. For each dataset, regardless of the size, we trained 10k iterations with batch size 32. All images are resized to 256x256. We used Adam solver [23] to update the model weights during training. In order to reduce model oscillation, we update the discriminators using a history of generated images rather than the ones produced by the latest generative models [47]. Similar to [59], we keep an image buffer that stores the 50 previously generated images. All networks were trained from scratch with a learning rate of 0.0002. Starting from 5k iteration, we linearly decay the learning rate over the remaining 5k iterations. For each dataset, the training takes about 1 day to finish on a single Titan X GPU. For qualitative evaluation, we compare the results with UNIT [31] and CycleGAN [59]. For quantitative evaluation, we tested unsupervised classification, and compare the results against standard benchmarks. In the end, we describe the 3DMM face shape prediction task as a specific

real-world application of our approach. Due to the simplicity of our model, our training takes significantly less memory and time comparing with CycleGAN and UNIT and can be trained with a more constrained budget: to train 200 epochs of the map dataset, each (approximately) requires: *12 hrs & 2.5 GB (ours); 36 hrs & 4.5 GB (CycleGAN); 44 hrs & 4.5 GB (UNIT).*

3.1 Qualitative Results

Figure 2 shows visual results of image translation on several datasets and compares with CycleGAN and UNIT. All models are trained using the same number of iterations, and the results are randomly sampled from the test set. Note that CycleGAN's results could be slightly different every time it trains, and the results shown are from a random trial. We can see from the examples that our results are sharper and of higher visual quality comparing with CycleGAN and UNIT; the colors of our results also better match with the inputs.

Figure 3 shows more of our image translation results in a wide range of settings. Although many of these tasks have very different source and target domains, our method shows strong generalization ability, demonstrating that our "perceptual similarity assumption" is general and widely applicable.

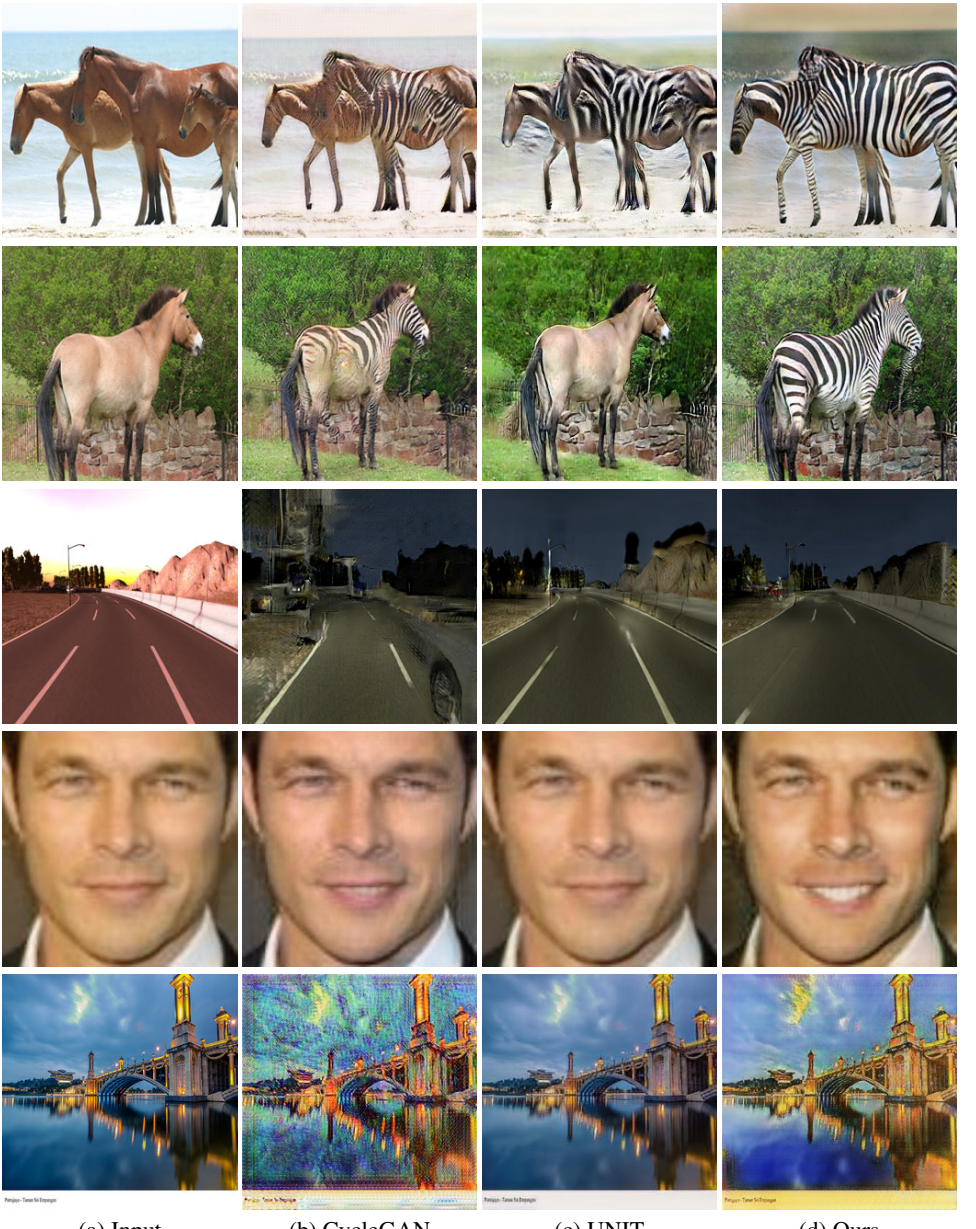
User study: To more rigorously evaluate the performance, we conduct a user study based on the image translation tasks. We asked for feedback from 10 users, by giving each user 30 tuples of images to rank. Each tuple contains results of CycleGAN, UNIT and our method on some translation task. The user study shows that our method achieves highest scores, outperforming CycleGAN and UNIT by a large margin: comparing with CycleGAN, ours are 64.4% (better), 20.0% (about the same) and 15.6% (worse); comparing with UNIT, ours are 91.7% (better), 7.1% (about the same) and 1.2% (worse).

Effects of using different layers as feature extractors: We experimented using different layers of VGG-19 as feature extractors to measure the perceptual loss. Fig. 4 shows visual example of the horse to zebra image translation results trained with different perceptual terms. We can see that only using high-level features as regularization leads to results that are almost identical to the input (Fig. 4 (c)) while only using low-level features as regularization leads to results that are blurry and noisy (Fig. 4 (b)). We find the balance by adopting the first three layers of VGG-19 as feature extractor which does a good job of image translation and also avoids introducing too many noise or artifacts (Fig. 4 (d)).

3.2 Quantitative Results

Map prediction: We translate images from satellite photos to maps with unpaired training data and compute the pixel accuracy of predicted maps. The original photo-map dataset consists of 1096 training pairs and 1098 testing pairs, where each pair contains a satellite photo and the corresponding map. To enable unsupervised learning, we take the 1096 photos from the training set and the 1098 maps from the test set, using them as the training data. At test time, we translate the test set photos to maps and again compute the accuracy. If the total RGB difference between the color of a pixel on the predicted map and that on the ground truth is larger than 12, we mark the pixel as wrong. Figure 5 and Table 1 show the visual results and the accuracy results, and we can see our approach achieves highest map prediction accuracy. Note that Pix2Pix is trained with paired data.

Unsupervised classification: We show unsupervised classification results on USPS [9] and MNIST-M [10] in Figure 6 and Table 2. On both tasks, we assume we have access to labeled MNIST dataset. We first train a generator that maps MNIST to USPS or MNIST-M and



(a) Input (b) CycleGAN (c) UNIT (d) Ours

Figure 2: Examples of image translation results comparing with UNIT and CycleGAN. From top to bottom: horse to zebra [19] 1, 2; dawn to night (SYNTHIA [42]); non-smile to smile (CelebA [32]), photo to Vangoh [19]. Best viwed in full resolution and color.

then use the translated image and original label to train the classifier. We can see from the results that we achieve the highest accuracy on both tasks, advancing state-of-the-art. The qualitative results clearly show that our MNIST-translated images both preserve the original

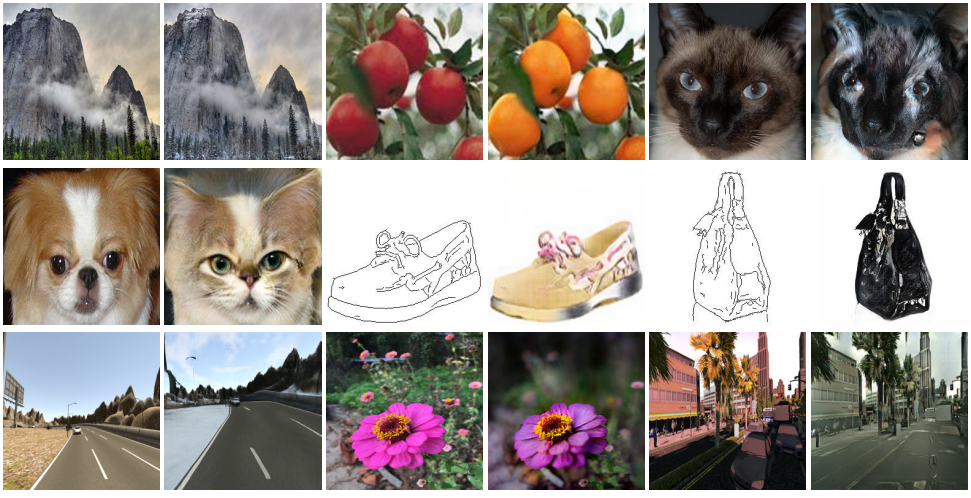
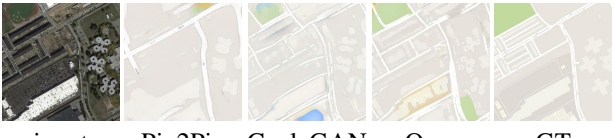


Figure 3: Examples of image translation results on a variety of datasets. From top to bottom and left to right: Yosemite summer to winter [19], apple to orange [19], cat to dog [38], dog to cat [38], edges to shoes [19], edges to handbags [19], SYNTHIA summer to winter [42], photo to DSLR [19], SYNTHIA to cityscape [8, 42].

(a) (b) (c) (d)

Figure 4: Effects of using different layers as feature extractors. From left to right: input (a), using the first two layers of VGG (b), using the last two layers of VGG (c) and using the first three layers of VGG (d).



input Pix2Pix CycleGAN Ours GT

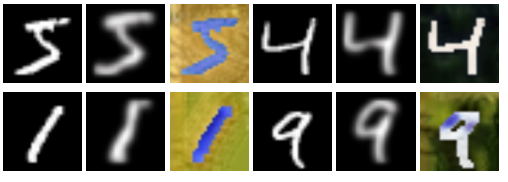
Figure 5: Unsupervised map prediction visualization.

Method	Accuracy
Pix2Pix [19]	43.18%
CycleGAN [59]	45.91%
Ours	**46.72%**

Table 1: Unsupervised map prediction accuracy.

label and are also visually similar to USPS/MNIST-M.

3DMM face shape prediction: As a real-world application of our approach, we study the problem of estimating 3D face shape, which is modeled with the 3D morphable model (3DMM) [3]. 3DMM is widely used for recognition and reconstruction. For a given face,

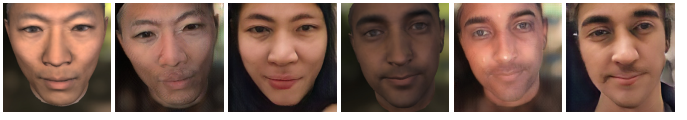


(a) (b) (c) (d) (e) (f)

Figure 6: Visualization of image translation from MNIST (a),(d) to USPS (b),(e) and MNIST-M (c),(f).

Method	USPS	MNIST-M
CoGAN [30]	95.65%	-
PixelDA [5]	95.90%	98.20%
UNIT [31]	95.97%	-
CycleGAN [59]	94.28%	93.15%
Target-only	96.50%	96.40%
Ours	**96.80%**	**98.33%**

Table 2: Unsupervised classification results.



(a) (b) (c) (d) (e) (f)

Figure 7: Visualization of rendered face to real face translation. (a)(d): input rendered faces; (b)(e): CycleGAN results; (c)(f): Our results.

Method	MSE
Baseline	2.26
CycleGAN [59]	2.04
Ours	**1.97**

Table 3: Unsupervised 3DMM prediction results (MSE).

the model encodes its shape with a 100 dimension vector. The goal of 3DMM regression is to predict the 100 dimension vector and we compare them with the ground truth using mean squared error (MSE). [50] proposes to train a very deep neural network [17] for 3DMM regression. However, in reality, the labeled training data for real faces are expensive to collect. We propose to use rendered faces instead, as their 3DMM parameters are readily available. We first rendered 200k faces as the source domain and use human selfie photo data of 645 face images we collected as the target domain. For test, we use our collected 112 3D-scanned faces as test data. For the purpose of domain adaptation, we first use our model to translate the rendered faces to real faces and use the results as the training data, assuming the 3DMM parameters stay unchanged. The 3DMM regression model structure is 102-layer Resnet [17] as in [50], and was trained with the translated faces. Figure 7 and Table 3 show the qualitative results and the final accuracy of 3DMM regression. From the visual results, we see that our translated face preserves the shape of the original rendered face and has higher quality than using CycleGAN. We also reduced the 3DMM regression error compared with baseline (where we trained on rendered faces and tested on real faces) and the CycleGAN results.

4 Conclusion

We propose an extremely simple yet effective model for image translation and domain adaptation and achieve superior performance in a variety of tasks demonstrated by both qualitative and quantitative evaluations. Our model is also more cost-effective, requiring significantly less training budget. Extensive experiments demonstrate that it is powerful and general, and can be easily applied to solve real-world tasks. As a result, we argue that image translation can be made simple whereas the commonly used cycle-consistency and shared latent space assumptions are not necessary.

References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[3] Volker Blanz, Sami Romdhani, and Thomas Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 202–207. IEEE, 2002.

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

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[6] Rui Caseiro, Joao F Henriques, Pedro Martins, and Jorge Batista. Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3846–3854, 2015.

[7] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 1, page 3, 2015.

[9] John S Denker, WR Gardner, Hans Peter Graf, Donnie Henderson, RE Howard, W Hubbard, Lawrence D Jackel, Henry S Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. In *Advances in neural information processing systems*, pages 323–331, 1989.

[10] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[11] 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.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 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*, pages 2672–2680, 2014.

[15] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

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

[21] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 746–753. IEEE, 2017.

[22] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)*, 33(4):149, 2014.

[26] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[27] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[29] Torgny Lindvall. *Lectures on the coupling method*. Courier Corporation, 2002.

[30] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[31] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[34] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[35] Aravindh Mahendran, Hakan Bilen, João F Henriques, and Andrea Vedaldi. Researchdoom and cocodoom: learning computer vision with games. *arXiv preprint arXiv:1610.02431*, 2016.

[36] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017.

[37] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[38] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[39] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *European Conference on Computer Vision*, pages 909–916. Springer, 2016.

[40] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, volume 2, 2014.

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

[42] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

[44] Andrei A Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*, 2016.

[45] 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*, pages 2234–2242, 2016.

[46] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 32(6):200, 2013.

[47] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016.

[50] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502. IEEE, 2017.

[51] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Xingchao Peng, Sergey Levine, Kate Saenko, and Trevor Darrell. Towards adapting deep visuomotor representations from simulated to real environments. *CoRR, abs/1511.07111*, 2015.

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

[53] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

[54] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[55] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[56] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. Pixel-level domain transfer. In *European Conference on Computer Vision*, pages 517–532. Springer, 2016.

[57] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv:1801.03924*, 2018.

[59] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[60] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017.