Universal Conditional Image Generator using Cycle-Consistent Adversarial Networks

Beakal Lemeneh
University of Rochester
Rochester, NY, USA
blemeneh@u.rochester.edu

Abstract

Zhu et al. *[19] introduced an unsupervised image-to-image translation technique from a source domain X to a target domain Y, where domain X and domain Y are unrelated and unpaired. It was able to achieve its goal by learning a mapping G : X → Y such that the distribution of images from G(X) was indistinguishable from the distribution Y using an adversarial loss which was then coupled with an inverse mapping F : Y → X to introduce cycle-consistency. Cycle consistency loss was introduced in the CycleGAN [19] model to enforce F(G(X)) ≈ X (and vice versa). I propose a new CycleGAN framework for generating multiple images that are indistinguishable from the images in domain Y conditioned on a single image of domian X, hence creating a universal image generator that enables sampling from the distribution $P(Y_{new} \mid x)$, where $x \in X$ and $Y_{new} \subset Y$, and Y_{new} is the set of possible images that could be created by combining different characteristics from the target domain.*

1. Introduction

Vincent Van Gogh and Claude Monet were some of the most prominent impressionist artists in the 19th century. Impressionist art revolved around accurately depicting objects' natural appearances, using dabs or strokes of primary unmixed colors to simulate the accurate depiction of light and the subject matter's primal essence. Since an impressionist painting accurately depicts a particular scenery or landscape, it is easy to imagine what the painting would look like in real life or what a color photograph could have documented if it were invented in the 19th century. Hence, it makes it a relatively easy task to do a one-to-one image to photo translation using CycleGAN [19]. Another question remains to be answered. What would the bank of the Seine near Argenteuil look like in the nighttime or on a cool summer, given that Claude Monet did that painting on a spring day in 1873? What would Van Gogh's famous painting, The Starry Night, depicting the city near Saint-Rémy-de-Provence, look like during the daytime? Implementation of the CycleGAN [19] framework lacks these features since it only does a one-to-one mapping between two unpaired domains, and outputs the most likely result, instead of outputting a domain of possible results.

In this paper, I present a more sophisticated version of CycleGAN that can learn to capture different interesting characteristics of one image collection and generate different possible combinations of how these characteristics are incorporated into the other image collection.

Given one set of images in domain X and a different set in domain Y, the model learns a mapping G : X → Y^N such that $y' \in y'^N$ from the output y'^N = G(x), $x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify y' apart from y.

2. Related Work

Generative Adversarial Networks (GANs) [4, 18] are frameworks created to successfully estimate generative models via an adversarial process. The adversarial loss incorporated in GANs forces the generated images to be indistinguishable from real images. This is done by simultaneously training two models: a generative model G that generates images from the data distribution. A discriminator model D estimates the probability of whether the images given to it came from G or the training sample. I adopt an adversarial loss to learn the mapping such that the translated images conditioned on some image from the source dataset are indistinguishable from the images in the target domain.

Unpaired Image-to-Image Translation Translating images from one domain X to domain Y (or vise versa) is an area that has been worked on for more than a decade. Rosales *et al*. [14] proposed a Bayesian technique that includes a prior based on a patch-based Markov random field computed from a source image and a likelihood term obtained from images with different rendering styles for inferring the

most likely target output.

CoGAN [12] was proposed for learning a joint distribution of multi-domain images by enforcing a weight-sharing constraint that limits the network capacity to incentivize a solution over a product of marginal distributions.

Liu *et al*. [11] builds upon the CoGAN [12] framework using the shared-latent space assumption to address the problem of inferring the joint distribution from the marginal distributions of images in different domain.

Huang *et al*. [6] proposed a framework where the translation of an image to another domain involves recombining its content code with a random style code sampled from the style space of the target domain.

CycleGAN [19] is another implementation that allows translation between two unpaired collection of images, which does not rely on any task-specific, predefined similarity function between the input and output, nor does it assume that the input and output have to lie in the same low-dimensional embedding space. I build my model on top CycleGAN architecture to generate multiple images conditioned on a single image from the source dataset.

Neural Style Transfer [2,3,8,16] is an image-to-image translation technique, where a content image is painted in the style of a style image (mostly a painting), based on matching the Gram matrix statistics of pre-trained features. The experiment done on this model has some elements of neural style transfer in it, where the content image dataset is a Monet painting, and the style image dataset is a random photo dataset. The generative networks and decoder (see Fig. 1) of this model are adopted from Johnson *et al*. [8]

3. Method

My goal is to learn the possible number of images that could be sampled from the domain Y conditioned on a single image from domain X given training samples $\{x_i\}_{i=1}^{M}$ where $x_i \in X$ and $\{y_t\}_{t=1}^{T}$ where $y_t \in Y$. I denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. As illustrated in Fig. 1, my model contains 'N' different mappings, G_j : X $\rightarrow$ Y where $j \in \{1, ..., N\}$, which are considered the adversarial generators in this model. They are in charge of generating 'N' images which are all different and conditioned on x, where $x \in X$. The mapping G : X $\rightarrow Y^N$ is the only mapping learned when in the model. The reverse mapping, F : Y $\rightarrow$ X, is also included in this model, but it is pre-trained on a vanilla CycleGAN [19] and only incorporated to G_0 (see Fig. 1) since the model is designed to only learn the mapping G : X $\rightarrow Y^N$. All the adversarial generative networks share a common adversarial discriminator network, D_y. D_y aims to distinguish between images found in Y and the generated images, $G_j(x)$. My objective contains three types of terms: adversarial losses [4] for matching the set of the generated images to the data distribution in the target domain, L_1 loss to add some variation in the generated images $G_j(x)$, and reconstruction loss to make the model forward cycle-consistent.

3.1. Adversarial Loss

Adversarial loss [4] is applied to all the mapping functions G_j : X $\rightarrow$ Y and their shared discriminator D_y, I express the objective as:

$$L_{GAN}(G_j, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log (1 - D_Y(G_j(x))] \quad (1)$$

where each G_j tries to generate image G_j(x) that look similar to images from domain Y , while D_y aims to distinguish between translated sample G_j(x) and real samples y. G_j aims to minimize this objective against an adversary D_y that tries to maximize it, i.e., $min_{G_j}\ max_{D_y}\ L_{GAN}(G_j,$ D_y , X, Y) . Since the mapping function F : Y $\rightarrow$ X and its discriminator D_x are pre-trained, their adversarial loss is not introduced in this model. The reasoning is that if the translated samples G_j(x) are forced to generate y, where Decoder(G_j(x)) $\approx$ y after introducing the reconstruction loss using the decoder (see Fig. 1), then the mapping F : Y $\rightarrow$ X, will, by definition, have an already minimized adversarial loss.

3.2. Reconstruction Loss

Zhu *et al*. [19] introduced cycle-consistency in his model to avoid training the same set of input images to any random permutation of images in the target domain. That is 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_j(x) $\rightarrow$ Decoder(G_j(x)) $\rightarrow$ F(Decoder(G_j(x))) $\approx$ x. Backward cycle consistency, where y $\rightarrow$ F(y) $\rightarrow$ G_j(F(y)) $\approx$ y, is implemented to this model only to the first generator G_0 as seen from the Fig. 1, since G_0 is the only mapping that is untrained in the model. Forcing a reconstruction loss [1] on G_j using a decoder preserves the forward cycle consistency of the model, since the mapping F : Y $\rightarrow$ X is pre-trained on vanilla CycleGAN [19]. Meaning, if Decoder(G_j(x)) $\approx$ y using the reconstruction loss [1], then x $\rightarrow$ G_j(x) $\rightarrow$ Decoder(G_j(x)) $\rightarrow$ y $\rightarrow$ F(y) $\approx$ x, which is the definition of forward cycle consistency [19]. The objective is described as:

$$L_{REC}(G_j) = \mathbb{E}_{x \sim p_{data}(x)}[|y - Decoder(G_j(x))|_1] \quad (2)$$

The reason reconstruction loss [1] is introduced in this model is, because pre-training F and minimizing reconstruction loss is more cheaper than enforcing cycle-consistency loss, since cycle-consistency involves learning both G and F. Besides, the objective of this model is to learn G, not F.

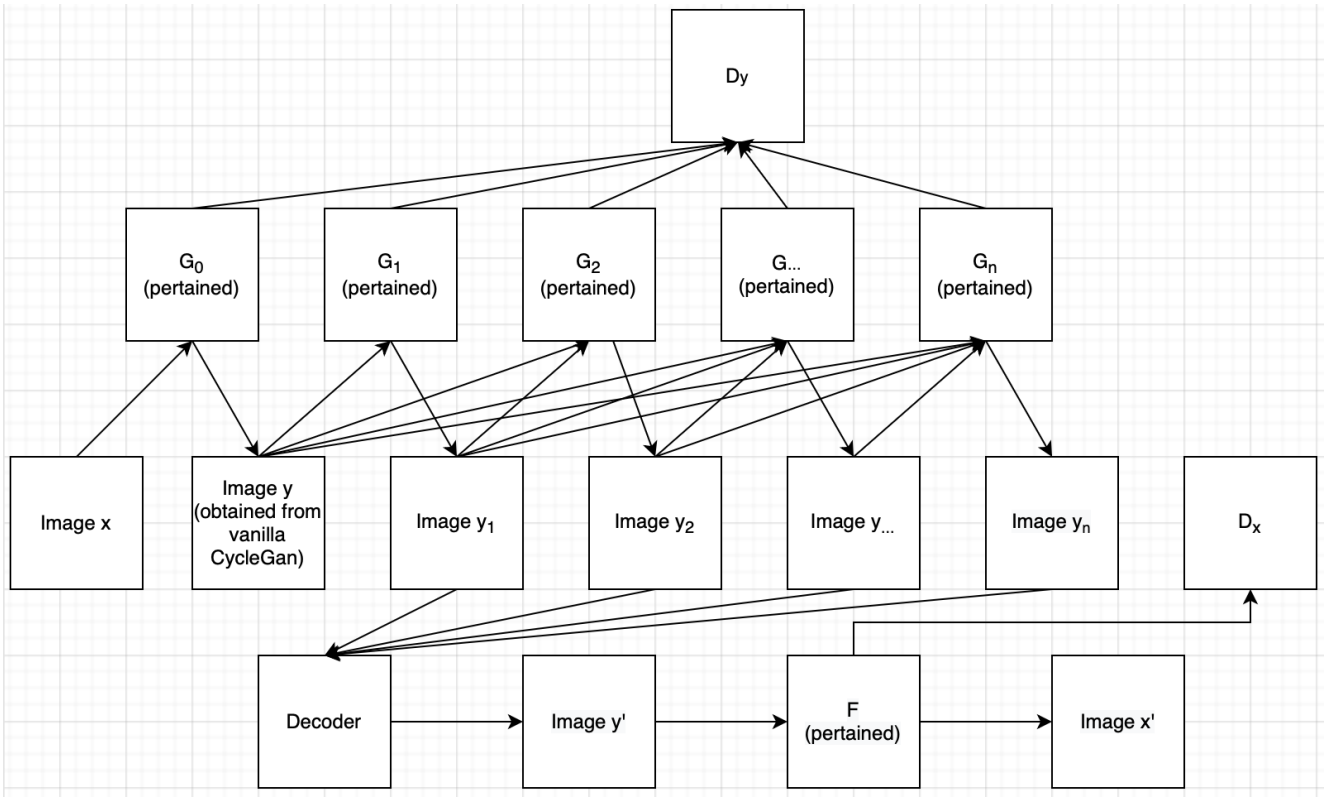

Figure 1. Architecture of the model that generates variational images.

3.3. Cycle Consistency Loss

Since the mapping, F : Y $\rightarrow$ X, and the discriminator D_x pre-trained on the vanilla CycleGAN [19] and the decoder forces a reconstruction loss [1] on the mapping G_j : X $\rightarrow$ Y, cycle consistency loss is preserved. I express the forward cycle consistency objective as:

$$L_{CYC}(G_j, F) = \mathbb{E}_{x\sim p_{data}(x)}[|F(Decoder(G_j(x))) - x|_1] \quad (3)$$

Since backward cycle consistency is implemented only to the first generator G_0 the backward cycle consistency objective is expressed as:

$$L_{CYC}(G_0, F) = \mathbb{E}_{y\sim p_{data}(y)}[|G_0(F(y)) - y|_1] \quad (4)$$

3.4. L1 Loss in the Set of Images Generated by G

Since the main objective of this model is to generate multiple images with different characteristics conditioned on some x, where $x \in X$, a variation needs to be added to $G_j(x)$. This is done by maximizing the L1 loss between the images generated by $G_j(x)$ and $G_k(x)$, for some $j \neq k$. I express the objective as:

$$L_{VAR}(G_j, G_k) = \mathbb{E}_{x\sim p_{data}(x)}[|G_j(x) - G_k(x)|_1], \quad where j \neq k \quad (5)$$

3.5. Full Objective

My full objective is:

$$L(G_j, D_Y, X, Y) = L_{GAN}(G_j, D_Y, X, Y) + L_{REC}(G_j) - \lambda_j L_{VAR}(G_j, G_k),$$
$$\text{where} \quad j \in \{1, ..., n\} \quad \text{and} \quad k = \{0, ..., j-1\} \quad (6)$$

where λ_j controls the relative importance of the variation of the image generated by G_j relative to the images generated by G_k, where k = {0,..., j-1}. I aim to solve:

$$G_{j^*} = \arg\min_{G_j} \max_{D_Y} L(G_j, D_Y, X, Y) \quad (7)$$

for all $j \in \{1, ..., n\}$

My model can be viewed as fine-tuning a recurrent pre-trained GAN [4, 15], where the each layer of GAN is sequentially generating images different from the previously

generated images. Another way of viewing my model is training an autoencoder [1] with multiple encoders generating different images on one hand, and a decoder reconstructing the images back to their original form, which is y, using the reconstruction loss.

4. Implementation

Network Architecture The architecture of this model is fully built on the CycleGAN [19] model. The generative networks are adopted from Johnson *et al*. [8]. Similar to Zhu *et al*. [19], two alternatives models are used to build the generative network and the decoder(see Fig. 1). The first architecture, ResNet [5], contains three convolutions, several residual blocks [5], two fractionally-strided convolutions with stride 12 , and one convolution that maps features to RGB which are all pre-trained on vanilla CycleGAN *et al*. [19].

As an alternative, a Unet [13] architecture with skip connections is also used to build the generative networks and the decoder. Similar to Johnson *et al*. [8], I use instance normalization [17] in both the ResNet [5] and Unet [13] with skip connections implementation. Similar to Zhu *et al*. [19], the discriminator networks uses 70 × 70 PatchGANs [7, 9, 10], which aim to classify whether 70 × 70 overlapping image patches are real or fake.

The reason behind pre-training all the generative networks is to avoid training all the generative networks from scratch, since they are all conditioned on a single image $x \in X$. Since they all have shared properties, pre-training them on a vanilla CycleGAN [19] and fine-tuning them to generate different images is a good choice for performance and efficiency.

Training Details The first generative network, G_0, is trained on two unrelated and unpaired datasets using vanilla CycleGAN [19]. G_0 is trained for 150 epochs on a learning rate of 0.0002 for the first 100 epochs and a linearly decaying learning rate from 0.0002 for the remaining 50 epochs. After G_0 is finished training, all the remaining G_j's are initialized to G_0. Masking technique is used to train the whole model. If G_j is being trained, then $\{G_k\}_{k=j+1}^{N}$ are frozen to let G_j finish its training process and generate G_j(x), since $\{G_k\}_{k=j+1}^{N}$ have to generate a different image from G_j. Hence, each generative network is trained sequentially. Each generative network is trained for 50 epochs on a learning rate of 0.0002. The result included in this paper is trained on a single layer GAN [4], G_1. I set $\lambda_1 = 1$ in Eq. (6). The loss function of this model tries to minimize the reconstruction loss in Eq. (2) and adversarial loss in Eq. (1) while maximizing the the L1 loss between each generative networks result in Eq. (5).

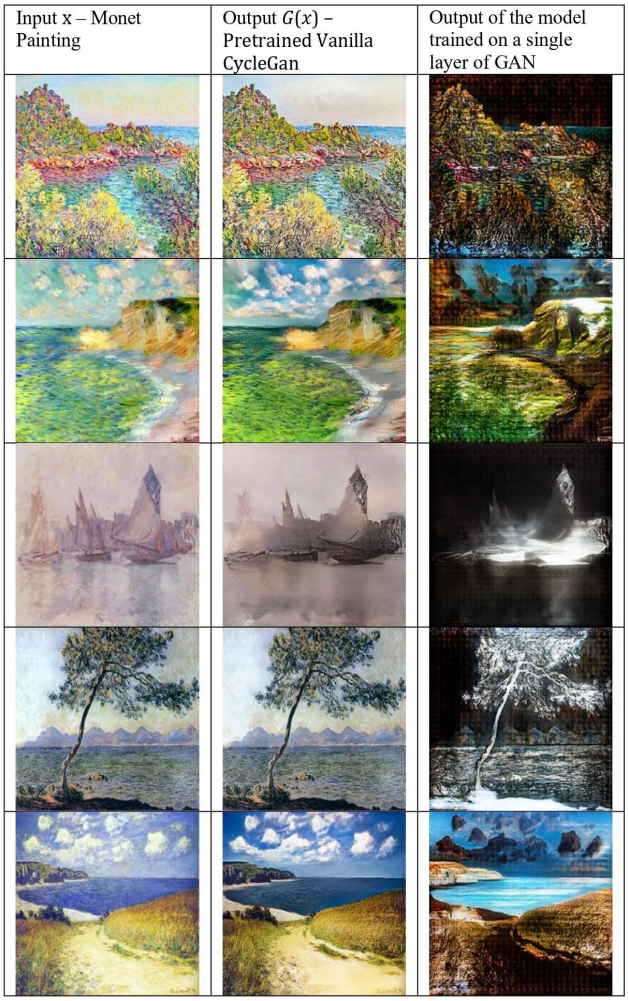

Figure 2. Result of the images of the model.

5. Experiments

This model has the same metric results as the CycleGAN [18] framework when compared to different unpaired image-to-image translation methods, hence the experimental results of this model on different evaluation metrics can be viewed at the CycleGAN paper.

The major difference that distinguishes this model from the CycleGAN [18] is its image generation techniques, which is demonstrated in Fig. 2. I demonstrate the generality of my algorithm on the monet2photo dataset. The PyTorch code can be found at my GitHub Account.

6. Conclusion

As described by Zhu *et al*. [19] and the results seen on Fig. 2, there is some issue of uniformity in the results. Although my method shows promising results, it is hard to

train, since it consists of training 'N' GANs [4] sequentially. On the other hand, incorporating an L1 loss between the images generated by the generative networks seems to get out of hand as the number of images that needs to be generated gets bigger.

Nonetheless, this paper makes the idea of universal image generation possible.

References

[1] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders, 2021. 2, 3, 4

[2] Leon A. Gatys, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Preserving color in neural artistic style transfer, 2016. 2

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 2

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 1, 2, 3, 4, 5

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[6] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation, 2018. 2

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 4

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016. 2, 4

[9] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017. 4

[10] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks, 2016. 4

[11] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks, 2018. 2

[12] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks, 2016. 2

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[14] Rómer Rosales, Kannan Achan, and Brendan J. Frey. Unsupervised image translation. *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 472–478 vol.1, 2003. 1

[15] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019. 3

[16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images, 2016. 2

[17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. 4

[18] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network, 2017. 1, 4

[19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. 1, 2, 3, 4