Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks: a Review

Anastasia Prisacaru

July 2020

1 Introduction

In this work we present an overview of the paper which was published by the Berkeley AI Research (BAIR) laboratory in 2017. In their paper [1], Jun-Yan Zhu et al. call their framework CycleGAN, as it is a combination of Generative Adversarial Networks (GANs) and the Cycle Consistency approach. The CycleGAN framework captures special characteristics of one image collection and figures out how these characteristics could be translated into other domains, all in the absence of any paired training examples. Figure 3 illustrates some examples of the CycleGAN approach, such as generating photos from paintings, changing textures of certain objects and transforming photos of summer landscapes into winter ones and vice versa.

This work is structured as follows: In Section 2, we introduce the image-to-image translation problem and previous approaches of solving it. Section 3 is focused on the CycleGAN approach, namely, its main components, implementation, training, evaluation and results. In Section 4, we discuss several applications of the CycleGAN approach, other than the ones presented in the paper. Finally, we conclude this work in Section 5.

2 Image-to-Image Translation

Image-to-image translation is the task of translating an image from one domain (e.g., painting), to another (e.g., photo). Ideally, other features of the image, such as the scene and the objects, should remain recognizably unchanged. There are dozens of research papers which use different approaches to perform image-to-image translation and many of them are quite successful. However, most of them use supervised learning, meaning that the model is trained on paired datasets, as shown on the left side of Figure 1. In this case, the model is trained on both the original data - the contour of the shoe - and the corresponding acquired image after the translation - the image of the shoe, which corresponds

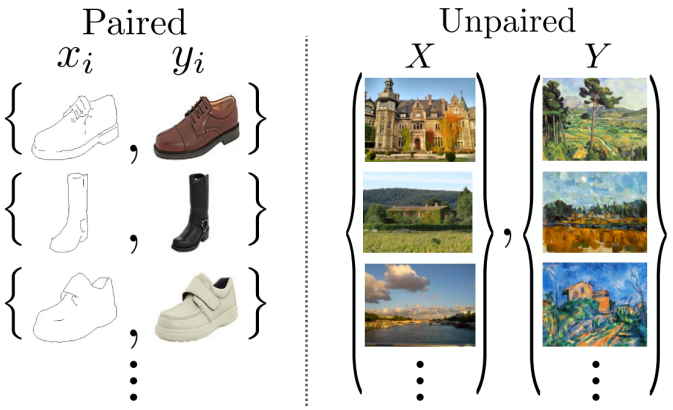

Figure 1: Examples of a paired and an unpaired dataset

to its contour. This kind of datasets proved to be perfectly suitable for image translation (e.g., pix2pix [2]), however they are difficult and expensive to obtain. Unpaired collections, on the other hand, are very easy to obtain, as the images from the source and target domains do not need to be related in any way. An unpaired dataset is illustrated on the right side of Figure 1, where the source domain contains a collection of photographs and the target domain - a collection of paintings of different landscapes.

3 The CycleGAN Approach

The CycleGAN approach, as the name suggests, consists of two parts: the cycle consistency and the Generative Adversarial Networks. Both of these terms are not new and have been used before, however, when combined together they create a novel and effective method of translating images to different domains. In the following subsections, we describe GANs, which are a popular and very efficient tool most commonly used in computer vision. Afterwards, we explain the Cycle Consistency approach and the way it enhances the performance of the GANs.

3.1 GAN

A GAN consists of two neural networks: a generator and a discriminator. A very comprehensive explanation of GANs can be found in the article of Thalles Silva [3]. In this paragraph, we focus on the image-to-painting example, where a photograph of a landscape should be translated into a Monet styled painting. We can imagine the generator as a painter, who tries to replicate Monet styled

paintings and the discriminator network - as an art expert, who must differentiate between real images and the fake ones created by the generator. These two networks compete against each other, however, they are also sending feedback to one another on how to get better. For instance, sometimes the art expert will tell the painter how he managed to detect that his painting was fake, so the painter would pay attention to that aspects, when painting that picture in the future. The generator, on the other hand, would share his tricks with the discriminator, which fooled him into thinking that the painting was real. Both networks are trained simultaneously until reaching equilibrium. In the perfect equilibrium, the generator would capture the general training data distribution. As a result, the discriminator would be always unsure of whether the generator's outputs are real or not.

3.2 Mode Collapse Problem

As mentioned before, the generator and the discriminator are neural networks which evolve simultaneously. Suppose that the generator has realized how the discriminator works. The discriminator has knowledge only about a collection of Monet paintings and no idea about the input dataset with photographs. Then the generator could start to cheat, by outputting the same image, which was previously classified by the discriminator as a Monet painting, because it would match the distribution of the target dataset. However, the output has nothing to do with the input image, because the generator produces the same picture every time. So, in other words, it is not guaranteed that there's any meaningful correspondence between the input and output, this problem being called Mode Collapse. In order to prevent this problem from happening, the authors combined GANs with the cycle consistency approach.

3.3 Cycle Consistency

A traditional example of the cycle consistency - is the translation from one language to another, which is often used by human translators. For instance, in order to check whether a translation of an English sentence into German was successful, the resulting German sentence would be translated back to English. If the resulting English sentence is the same as the original one, then the German translation is proved to be successful. Following the same intuition, the authors added another GAN to the transformation and as a result got 2 generators (G and F) and 2 discriminators (D_x and D_y), as shown on the left side of Figure 2. The generator G generates a painting and F takes this image and tries to translate it back to domain X. Hopefully, the final generated image is the same as the original one.

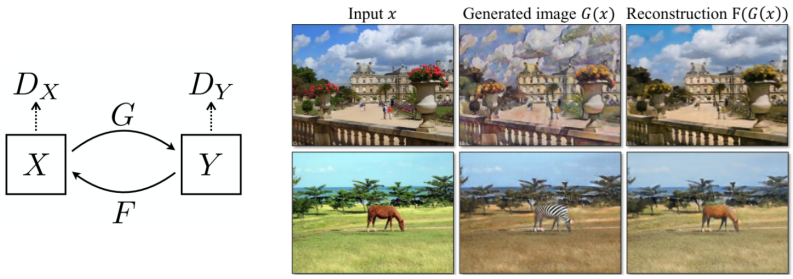

Figure 2: Cycle Consistency

3.4 Loss function

The network is trained using multiple losses which are combined together, how it is shown in Formula 1. The authors use adversarial losses of the two generator-discriminator pairs F and G, just like a general GAN. However, they also use a cyclic loss, which is nothing more than an additional GAN, which proves whether the generated image matches the original one. Based on the images in Figure 2, we can identify the original image, the generated one and the reconstructed one. The cyclic loss has a lambda parameter, which scales the importance of the two objectives.

$$\mathcal{L}(G, F, D_x, D_y) = \mathcal{L}(G, D_y, X, Y) + \mathcal{L}(F, D_x, Y, X) + \lambda \mathcal{L}_c yc(G, F) \quad (1)$$

3.5 Implementation

The generator has 3 main parts: encoder, transformer and decoder. The encoder is a set of 3 convolution layers. It takes an image as input and outputs a feature map. The transformer takes this feature map and parses it through 6 residual blocks. Each residual block is a set of 2 convolution layers and a bypass. This bypass allows the transformations of earlier layers to be retained throughout the network and therefore allowing building a deeper neural network. The decoder is the exact opposite of the encoder. It takes the transformer's output, which is another feature map and outputs a generated image. This is done with 2 deconvolution layers to rebuild from the low-level extracted features and upscale the image. Then a final convolution layer is applied to get the final generated image. The output image is passed on to the discriminator.

The discriminator takes the output of the generator and decides whether it is a part of the real dataset or the fake generated image dataset. This architecture is a patch GAN [4]. It involves chopping an image into 70 cross 70 overlapping patches, running a regular discriminator over each patch and averaging the results.

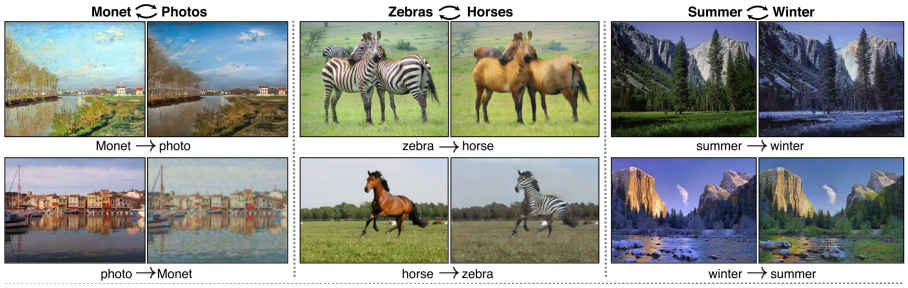

Figure 3: Examples of applications of the CycleGAN approach, from left to right: generating photos from paintings, object transfiguration and season translation

3.6 Training

To prevent model oscillation as well as model overfitting, the authors stored the last 50 generated images and fed them to the discriminators, rather than just the one image produced by the generators. The Adam algorithm, which stands for adaptive moment estimation, with a batch size of 1 was used for optimizing the training process. The learning rate was set to 0.0002 for the first 100 epochs, and then linearly reduced to zero over the next 100 epochs. The least-squares loss was used instead of the log likelihood objective, as it is more stable during training and generates higher quality results. The lambda parameter was set to 10 for all experiments.

3.7 Evaluation

The CycleGAN approach was compared with other image-to-image translating baselines, which use variations of GANs trained on unpaired datasets, namely: BiGAN [5], CoGAN [6], feature loss GAN [7], SimGAN [7]. They have also compared CycleGAN against the pix2pix [2] approach, in order to observe how close they got to the "upper bound" without using any paired training data. The authors used two different evaluation metrics: AMT perceptual studies and FCN score with semantic segmentation metrics.

AMT stands for Amazon Mechanical Turk, which is a crowd-sourcing website for businesses to hire remotely located crowd-workers. The authors performed AMT perceptual studies on the map→aerial photo task, where participants were shown a sequence of pairs of images, one - a real photo or map and the other one - a fake photo (generated by CycleGAN or a baseline). As a result, CycleGAN outperformed the other baselines and the generated pictures fooled participants on around a quarter of trials, in both the maps→aerial and the aerial→photos translations.

Figure 4: Examples of failed attempts

Additionally, the authors adopted the "FCN score" metric, in order to evaluate the model automatically, without involving human experiments. FCN is a metric which stands for a fully-convolutional network and it predicts a label map for a generated photo. This label map was then compared against the input ground truth labels using standard semantic segmentation metrics. The authors used the standard metrics from the Cityscapes benchmark [8], including per-pixel accuracy, per-class accuracy, and mean class Intersection-Over-Union, in order to evaluate the performance of the photo→label task. CycleGAN outperformed the baselines, which were trained on unpaired datasets. However, it is not as good as the pix2pix approach.

3.8 Results

The CycleGAN approach has shown good results on the following tasks: paintings ↔photos, zebra↔horses, winter↔summer, aerial↔map and photo enhancement. The first three examples are shown in Figure 3. This approach has also shown promising results on generating videos for the day↔night [9] and zebra↔horse [10] examples. However, it has limitations, a few examples of which can be observed in Figure 4. Tasks that require substantial geometric changes, such as cat-to-dog or orange-to-apple translations, usually fail. Furthermore, some failure cases are caused by the distribution characteristics of the training datasets. For example, the horse→zebra example on the right side of Figure 4, has got confused, because the model was trained on images of wild horses and zebras, which do not contain images of a person riding them.

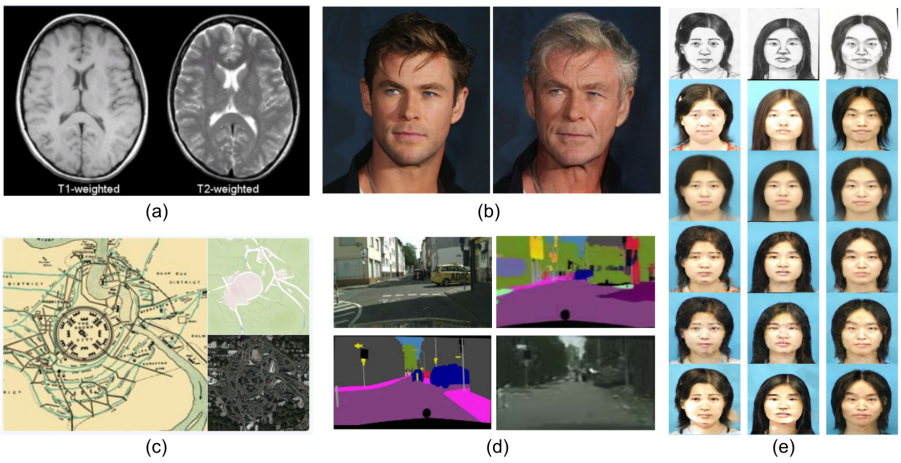

Figure 5: Other applications of the CycleGAN approach: (a) Medical Image Synthesis [11]; (b) Face-aging [12]; (c) Resurrecting Ancient Cities [13]; (d) Image Segmentation [1]; (e) Sketch-to-Photo Synthesis [14].

4 Other Applications of CycleGAN

The CycleGAN approach has also been applied for various practical tasks, such as:

- Medical Image Synthesis [11]: the CycleGAN approach can successfully transform MRI sequences from T1-weighted to T2-weighted scans and vice versa, as shown in Figure 5 (a).
- Face-aging [12]: The face aging app FaceApp, which is based on CycleGAN, went viral across the world. The app can do forward aging from 20s to 50s (shown shown in Figure 5 (b)) and reverse aging from 50s to 20s.
- Resurrecting Ancient Cities [13]: convert ancient maps of Babylon (see Figure 5 (c)), Jerusalem and London into modern Google Maps and satellite views.
- Image Segmentation [1], as shown in Figure 5 (d), has the potential to be used for autonomous driving.
- Sketch-to-photo synthesis [14]: as shown in Figure 5 (e), the CycleGAN framework can generate various faces from a single sketch, which has a big potential for being used in by the police.
- Cryptography [15]: the CipherGAN framework, which is based on CycleGAN, is able to solve shift ciphers with high accuracy.

- Converting Fortnite into PUBG [16]: the CycleGAN framework can translate between two popular Battle Royale games Fortnite and PUBG, so that users can enjoy playing one game in the visuals of the other one.

5 Conclusion

The CycleGAN approach proved to be effective in image-to-image translation tasks, where paired data is hard to find. It works well on tasks which involve color or texture changes, like photo↔painting, day↔night, collection style transfer, photo translations and photo enhancement, however it has its limitations. Once the task involves geometric changes or more complex object segmentation (as in the example of a human riding a horse) it fails to perform the transformation. The CycleGAN framework outperforms other baselines which are trained on unpaired datasets and achieve very good results, comparable to the quality to the fully supervised pix2pix approach. Since paired data is harder to find in most domains, and not even possible in some, the unsupervised training capabilities of CycleGAN are quite useful and can be applied in various real-world applications, such as: face generation for police data, medical data generation, face aging, image segmentation for self driving cars and the resurrecting of ancient cities.

References

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[3] Thalles Silva. An intuitive introduction to generative adversarial networks (gans). FreeCodeCamp `https://shorturl.at/ahoJU`, 2018. Accessed: 2020-07-12.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

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

[7] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[9] parktaesung89. [cyclegan] rendering day driving in night style. `https://www.youtube.com/watch?v=N7KbfWodXJE`, 2017. Accessed: 2020-07-20.

[10] Jun-Yan Zhu. Turning a horse video into a zebra video (by cyclegan). `https://www.youtube.com/watch?v=9reHvktowLY`, 2017. Accessed: 2020-07-20.

[11] Siddhartha Mishra. Medical image synthesis using cyclegan (mri t1w to t2w). Analytics Vidhya `https://shorturl.at/itKXZ`, 2019. Accessed: 2020-07-20.

[12] Anil Chandra. Implementing cyclegan for age conversion. `https://blog.paperspace.com/use-cyclegan-age-conversion-keras-python/`, 2015. Accessed: 2020-07-20.

[13] Jack Clark. Import ai: Issue 45: Starcraft rumblings, resurrecting ancient cities with cyclegan, and microsoft's imitation data release. `https://jack-clark.net/2017/06/05/import-ai-issue-45/`, 2017. Accessed: 2020-07-20.

[14] Lidan Wang, Vishwanath Sindagi, and Vishal Patel. High-quality facial photo-sketch synthesis using multi-adversarial networks. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 83–90. IEEE, 2018.

[15] Aidan N Gomez, Sicong Huang, Ivan Zhang, Bryan M Li, Muhammad Osama, and Lukasz Kaiser. Unsupervised cipher cracking using discrete gans. *arXiv preprint arXiv:1801.04883*, 2018.

[16] Chintan Trivedi. Turning fortnite into pubg with deep learning (cyclegan). Towards Data Science `https://shorturl.at/cgiqM`, 2018. Accessed: 2020-07-20.