# Exploring: "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks"

**Authors**

**KTH Royal Institute of Technology**
**Altan Senel**
altans@kth.se

**KTH Royal Institute of Technology**
**Ayberk Uckun**
uckun@kth.se

## Abstract

In recent years, generative adversarial networks have been achieving impressive results, especially in image-to-image translation problems. However, paired datasets are non-existent for many of the tasks, making these networks unattractive to train. CycleGAN successfully addresses this problem by introducing cycle consistency loss. We implement the architecture and the training from scratch and replicate some experiments with the aim of having a deeper understanding of CycleGANs and generative adversarial networks.

## 1 Introduction

The image-to-image translation is a problem in computer vision of learning the mapping between an input image domain $A$ and output image domain $B$ [1]. Works on this domain are mostly done by using paired datasets. However, finding paired datasets for two arbitrary image domains is challenging. By paired dataset, we mean that we would need paired images of a cat and a dog that would look exactly similar in their positioning, lighting, background etc.

This is where CycleGAN, which is a generative adversarial network, excels with learning these mappings from domain $A$ to $B$ with unpaired datasets [1]. Generative adversarial networks are first introduced in 2014, which the network consists of a generative model $G$ and a discriminator model $D$ which estimates whether a sample came from the real training data or was generated by $G$ [2]. Generator and discriminator compete against each other during the learning, hence why these networks are called adversarial. For CycleGAN, there are two generator and discriminator pairs for translations from domain $A$ to $B$ and for the reverse direction. Also in order to ensure proper translation between domains a cycle consistency loss is introduced in CycleGAN for it to allow unpaired image-to-image translation [1].

Firstly, we tried to understand the architecture in depth and tried implementing CycleGAN from scratch by only using the CycleGAN paper and no supplementary material. However, the paper was not explicit in some parts of the architecture, for example, the spatial padding. After the implementation, we tried to recreate the results for a subset of the experiments in the paper and performed qualitative comparisons with the results from the paper.

## 2 Related Work

Previously, generative adversarial networks are applied to many challenging vision problems such as realistic image manipulation with user-controlled queries [3], automatic synthesis of realistic images from text [4], predicting future images from a video sequence [5] and many more. Similarly,

CycleGAN builds on the concept of adversarial loss introduced in [2]. Adversarial loss in the image-to-image translation context means optimizing the parameters of the generator so that the translated image is indistinguishable from a real image. Discriminators' job is to be able to tell the difference between a translated image and a real image.

A recent approach in image-to-image translation tackles the pixel-wise image segmentation prediction problem, which can be formulated as an image translation problem, employs fully convolutional networks [6]. CycleGAN builds upon the "pix2pix" framework, which uses a conditional generative adversarial network architecture and a learned loss function instead of a hand-engineered one [7]. The main advantage of CycleGAN over the pix2pix framework is that it does not require a paired dataset.

Previous works on unpaired image-to-image translation include the bayesian approach for inferring the most likely output image [8]. A more similar approach employs a weight-sharing constraint that favours learning a joint distribution of multi-domain images over the product of marginals [9]. Different from many other approaches, CycleGAN does not rely on task-specific similarity functions or that input and output should lie in the same low dimensional space [1]. With fewer assumptions, CycleGAN becomes more of a general purpose architecture for different computer vision problems.

# 3 Data

We used four different datasets and applied the same preprocessing to all of them.

## 3.1 Datasets

### 3.1.1 Horse2Zebra and Apple2Orange

These two datasets consist of images from ImageNet [10]. We downloaded these datasets using the "cycle_gan" collection of TensorFlow Datasets [11] and Tensorflow provides these datasets using the published links for the datasets by the authors of [1]. Authors created these datasets using the keywords "wild horse", "zebra", "apple", and "navel orange" and scaled the images to 256x256 resolution. The resulting datasets had 989 horse, 1177 zebra images and 996 apple, 1020 orange images.

### 3.1.2 Summer2Winter Yosemite

This dataset consists of photos taken in Yosemite National Park and uploaded to Flickr. We accessed the dataset using the UC Berkeley's official directory for CycleGAN Datasets [12]. The authors retrieved the images using the Flickr API with the tag "yosemite" and "data-taken" field, pruned the black and white images and scaled the resolution to 256x256. The resulting dataset had 1273 summer and 854 winter images.

### 3.1.3 Cityscapes

Cityscapes dataset contains 5000 fine pixel-level annotated and 20k coarse annotated street scenes from 50 different cities taken from recorded video sequences [13]. The authors took 2975 fine annotated images and their raw images as a training dataset and used 500 images of the validation set as the test set. We accessed this dataset using Tensorflow Datasets. Images have 128x128 resolution.

## 3.2 Preprocessing

Preprocessing includes jittering and normalizing an image but not flipping. To normalize an image, we brought images to the range of [-1, 1] and to jitter, we re-scaled them to 286x286 resolution using the Nearest Neighbor method and randomly cropped the image back to 256x256. This way we augment the images to avoid overfitting.

# 4 Methods

## 4.1 Adversarial Loss

The purpose of unpaired image-to-image translation is to learn a mapping function between two domains. CycleGAN has two mapping functions G: X $\rightarrow$ Y and F: Y $\rightarrow$ X and also two adversarial discriminators. G tries to match the generated images' distribution with the target distribution while $D_y$ discriminates between real and fake domain images. These two loss components create an adversarial min-max loss. While the original adversarial loss is consist of the negative log-likelihood, to better stabilize the training, the authors applied least-square loss following the work [14]. Overall adversarial loss for the Y domain is given in Equation 1. The first two components of the loss's right-hand side belong to the discriminator while the third one belongs to the generator. Interchanging the places of x and y gives other domain's loss.

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

## 4.2 Cyclic Loss

In theory, any mapping from domain X to Y is correct as long as it matches the target distribution. Thus, adversarial loss alone does not guarantee a proper image translation. To ensure that only the required parts of the image change and the image does not lose its authenticity, one must decrease the number of possible mapping functions by adding constraints. CycleGAN argues that by ensuring cycle consistency between domains, that is, a translated image by G can be easily translated back to its original domain by F, we can learn the proper mapping functions. So CycleGAN introduces Equation 2 as an additional loss to generators.

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

## 4.3 Identity Loss

Identity Loss is suggested by [15] and used in CycleGAN for some datasets when required. This additional loss is added to regularize generators to preserve the original colour of the image. Normally changing the colour is a valid approach that a generator can apply while, for example, generating summer-to-winter images. However, in the case of photos-to-paintings, preservation of the colour is important. This loss is applied in Equation 3 by giving X domain's image to that domain's generator and expecting the output to be the same.

$$\begin{aligned}\mathcal{L}_{\text{identity}}\left(G, F\right) = \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\|G(y) - y\|_1\right] \\ + \mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\|F(x) - x\|_1\right]\end{aligned} \tag{3}$$

## 4.4 Instance Normalization

Instance normalization, also known as contrast normalization, is an adaptation of batch normalization for generative networks proposed by [16]. It can be seen as equal to the batch normalization when the batch size is 1 but unlike batch normalization, instance normalization is applied also in test time. This kind of normalization simplifies the generation process by removing the instance-specific contrast information from the image.

## 4.5 Reducing Model Oscillation

Oscillation of the loss is a common occurrence while training Generative networks. To make the discriminator more stable for a better training, [17] suggests using the collection of generated images instead of using the latest one as a strategy. CycleGAN uses the latest 50 generated images as an image pool to draw a batch size of images to calculate the discriminator loss. This way discriminator can continue learning its mistakes from the previous images while adapting to new ones.

### 4.6 PatchGAN

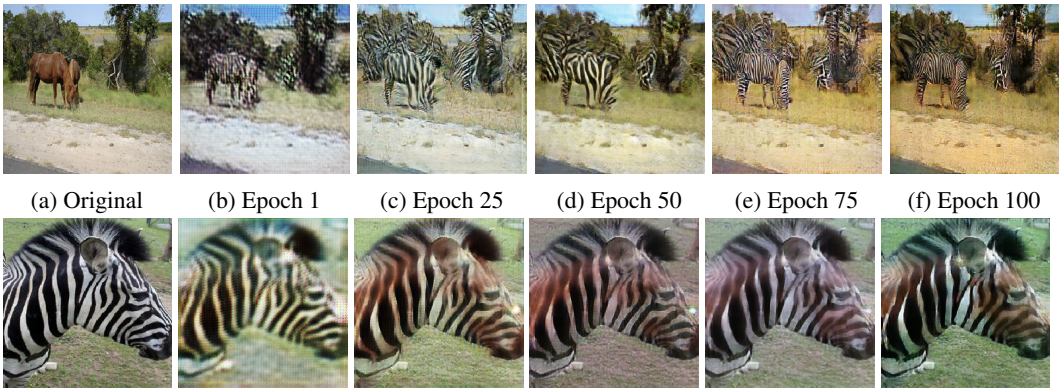
For the discriminator of the cycleGAN, 70x70 PatchGANs [7] are used. These GANs are designed to classify overlapping 70x70 patches of an image. This way discriminator has fewer parameters, runs faster and can work on arbitrary size images. Decreasing the patch size up to the level of PixelGAN only decreases the quality (makes it blurry) while increasing up to ImageGAN level only makes it harder to train. A PatchGAN discriminator models the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter. [7]

### 4.7 Upsampling the checkerboard effect

In the generator, after a series of down-sampling operations using 2d convolutions, upsampling is done using deconvolutions to obtain the original spatial dimensions. Our deconvolution kernel is of size 3 and stride is 2, therefore, there is an uneven overlap when sliding these kernels. This effect is called the checkerboard effect or in some cases, mosaic pattern [18]. This effect is especially observed in the early samples generated by the generator. For future work, new forms of regularization or parameter sharing can be employed as suggested in [18]. Alternatively, all deconvolutions can be replaced with a different upsampling technique that is not susceptible to the checkerboard effect.

## 5 Experiments

### 5.1 Horse2Zebra



(a) Original (b) Epoch 1 (c) Epoch 25 (d) Epoch 50 (e) Epoch 75 (f) Epoch 100

(g) Original (h) Epoch 1 (i) Epoch 25 (j) Epoch 50 (k) Epoch 75 (l) Epoch 100

Figure 1: Generated Images using Horse2Zebra Dataset. First row horse-to-zebra, second row zebra-to-horse

Figure 1, shows the generated images throughout the training of CycleGAN on Horse2Zebra dataset. We can see that epoch 1 images are all blurry and low quality but it shows a little bit of the characteristics of the target domain, like black and white lines for zebras and brown colour for a horse. On epoch 25, it becomes clearer that the network learned that it should apply these transformations more commonly and also network produces higher quality images. On epoch 75, the network not only learned that it should change the texture and the colour of the animal but also the background since horses generally live around greener grass compared to zebras living in savannas. On epoch 100, it starts solving the problem of mixing the background tree with the horse itself and also produces better quality images.

We train our models for 100 epochs compared to the original experiments with 200 epochs and also we use mixed precision in calculations because of resource constraints. Our images are not perfect but they show clear signs of learning and somewhat successful image translation. The authors also warn about the existence of failure cases. It is possible to find better-transformed images and worse-transformed images in the dataset thus showing the results turns into a cherry-picking process since there is no proper consistency and a metric to measure the overall performance of the network.

We can only visually evaluate these unpaired images and this is what the authors did in the CycleGAN paper using "Mechanical Turks" and by asking real or fake.

### 5.2 Apple2Orange

Figure 2: Original apple image

Figure 3: Generated orange image after 20 epochs

Figure 4: Original orange image

Figure 5: Generated apple image after 20 epochs

Out of the image-to-image translations tackled in this report, the most trivial mapping seems to be in the case of apple2orange. With only colour and texture changes, it is possible to make an apple look like an orange and vice-versa. This task does not require learning geometric transformations of the objects, therefore with fewer epochs, decent results were achieved by generators.

Colour transformation is learned really early on in the training process. With more epochs, generators learn to remove artefacts such as the checkerboard effect and make the images sharper, which will complicate the job of the discriminator.

### 5.3 Summer2Winter Yosemite

For this dataset, again, we see somewhat more successful results compared to Horse2Zebra dataset since still only a colour change somewhat can solve the problem. Over the course of the training, the network learns that generated summer images must be greener and brighter while winter images must be whiter and darker. The translation from Figure 6a to Figure 6f and from Figure 6m to Figure 6r clearly shows this phenomenon.

Moreover, in Figure 7, we can see a generated winter image using a dark blue and orange colour plate which suggests a more nightly autumn image. This kind of image generation can happen since we are not using identity loss and these kinds of images fit more to the distribution of winter images rather than summer images.

Loss graphs of the CycleGAN while training ON Summer2Winter Yosemite are given in Figure 8. Losses are smoothed using 0.99 exponential moving averaging yet the real graph can be seen in the background. The original loss graph has lots of oscillation since the loss values are reported after every 5 batches with batch size 1. Also, the oscillating loss graphs are common in adversarial learning since the generator and the discriminator battle against each other and when one of them is doing better this means the other one is doing worse hence oscillating the loss. We can see that the biggest contributor to the generator loss is the cyclic loss and it decreases steadily. Decreasing its value lowers the total generator loss even though the adversarial loss of the generator slightly increases. Also for the discriminators, we can see that their loss generally stays stable after a quick drop.

### 5.4 Cityscapes

The original image is the ground truth pixel-wise segmentation of a street image in this experiment. There are two generators that are learning the mapping from pixel-wise segmentation to street photography and reverse mapping. Here we specifically look at the images generated by the generator that creates street photography.

As the result of deconvolution with a kernel size of 3 and stride of 2, we can clearly see the checkerboard effect in the generated image after 1 epoch of training. The image is quite blurry and there are not many details. After 50 epochs of training, the generator got better at generating realistic images and the checkerboard effect is minimized. The image is less blurry, however, a good

(a) Original (b) Epoch 1 (c) Epoch 25 (d) Epoch 50 (e) Epoch 75 (f) Epoch 100

(g) Original (h) Epoch 1 (i) Epoch 25 (j) Epoch 50 (k) Epoch 75 (l) Epoch 100

(m) Original (n) Epoch 1 (o) Epoch 25 (p) Epoch 50 (q) Epoch 75 (r) Epoch 100

(s) Original (t) Epoch 1 (u) Epoch 25 (v) Epoch 50 (w) Epoch 75 (x) Epoch 100

Figure 6: Generated Images using Summer2Winter Yosemite Dataset. The first two rows are summer-to-winter, third and fourth rows are winter-to-summer

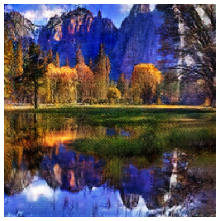


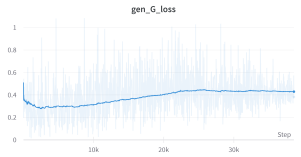
Figure 7: Generated relatively dark blue summer-to-winter image

discriminator should still be able to determine if it is generated and not sampled from a real dataset. The generated image after 100 epochs of training is much more realistic. The road and the trees look somewhat realistic.

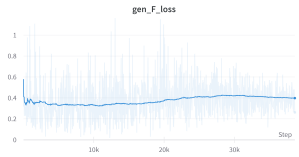
Cyclegan is incentivized to create sharp images, as greyish blurred outputs can be identified by the discriminator network [1]. In this experiment, we clearly see that the generated image got sharper with more training hence getting closer to the true colour distribution.

## 6 Conclusion

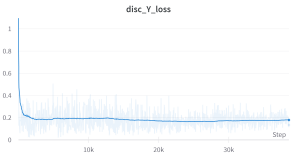
We learned that image-to-image translation is an achievable task even without the existence of paired datasets by learning the domain distributions and proper well-constrained mapping functions between these distributions. Cyclic loss, identity loss and several other ideas implemented in the CycleGAN paper show the way of finding solutions to these kinds of problems. Even though CycleGAN is successful with translations that include colour and texture changes, it fails with the ones that



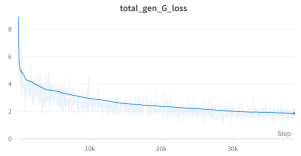

(a) Cyclic Loss




(b) Adversarial Loss of Generator G

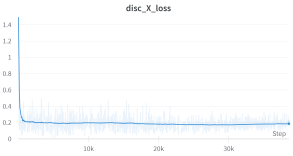



(c) Discriminator Y Loss

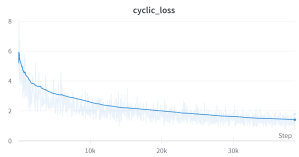



(d) Adversarial Loss of Generator F


(e) Discriminator X Loss




(f) Total Loss of Generator G




(g) Total Loss of Generator F

Figure 8: Loss Graphs of CycleGAN Training

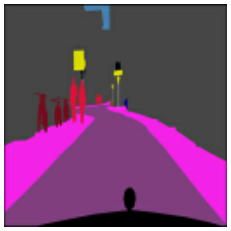


Figure 9: Original segmentation image

Figure 10: Generated city image after 1 epoch

Figure 11: Generated city image after 50 epochs

Figure 12: Generated city image after 100 epochs

require geometric changes. This requires further exploration. Moreover, the non-existence of proper quantitative methods to measure the performance of the network on unpaired domain translation tasks is a problem which must be considered to create a proper baseline for this task.

We also learned that training GANs is a challenging, resource-hungry process. We were able to reproduce experiments to a reasonable extent, however, this challenge also taught us the ways how one can efficiently process data and write faster training functions. Implementation from scratch was a more challenging task than we anticipated thus while we are experimenting with datasets, we also needed to experiment with the network architecture itself, ie. unspecified ResNet block, left out activation, ambiguous padding and we needed to experiment with the differentiation library we use since differentiation of multiple models under the same custom loop, implementing extra image pooling components and trying to do all these in a faster fashion by applying several tweaks and by using graph functions for faster execution is not an easy task. Also, an artefact that was discovered during experimentation was the checkerboard effect, which was the result of our deconvolutions in the generative model. Instead of the deconvolutions mentioned in the paper, for future work, different upsampling techniques can be experimented with to get rid of the checkerboard effect.

You can find the relevant written-from-scratch code, network architecture and implementation details in our GitHub repository: `https://github.com/ayberkuckun/DeepLearningDataScienceProject`

# 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*. 2017, pp. 2223–2232.

[2] 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* 27 (2014).

[3] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. "Generative visual manipulation on the natural image manifold". In: *European conference on computer vision*. Springer. 2016, pp. 597–613.

[4] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. "Generative adversarial text to image synthesis". In: *International conference on machine learning*. PMLR. 2016, pp. 1060–1069.

[5] Michael Mathieu, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error". In: *arXiv preprint arXiv:1511.05440* (2015).

[6] 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*. 2015, pp. 3431–3440.

[7] 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*. 2017, pp. 1125–1134.

[8] Rómer Rosales, Kannan Achan, and Brendan J Frey. "Unsupervised image translation." In: *iccv*. 2003, pp. 472–478.

[9] Ming-Yu Liu and Oncel Tuzel. "Coupled generative adversarial networks". In: *Advances in neural information processing systems* 29 (2016).

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[11] *Cycle_gan: tensorflow datasets*. URL: https://www.tensorflow.org/datasets/catalog/cycle_gan.

[12] *Index of /taesung_park/CycleGAN/datasets*. URL: https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/.

[13] 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: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[14] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. "Least Squares Generative Adversarial Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2813–2821. DOI: 10.1109/ICCV.2017.304.

[15] Yaniv Taigman, Adam Polyak, and Lior Wolf. *Unsupervised Cross-Domain Image Generation*. 2016. DOI: 10.48550/ARXIV.1611.02200. URL: https://arxiv.org/abs/1611.02200.

[16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization*. 2016. DOI: 10.48550/ARXIV.1607.08022. URL: https://arxiv.org/abs/1607.08022.

[17] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. *Learning from Simulated and Unsupervised Images through Adversarial Training*. 2016. DOI: 10.48550/ARXIV.1612.07828. URL: https://arxiv.org/abs/1612.07828.

[18] Jon Gauthier. "Conditional generative adversarial nets for convolutional face generation". In: *Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester* 2014.5 (2014), p. 2.