Image translation using generative adversarial networks (GANs)

Reda Bouaida
ENS Paris-Saclay
reda.bouaida@ens-paris-saclay.fr

Elmoustapha Ebnou
ENS Paris-Saclay
elmoustapha.ebnou@ens-paris-saclay.fr

Abstract

This paper reviews current state of the art methods for image-to-image translation. The goal here is to learn a mapping between an input image in domain X and an output image in a domain Y in the unpaired/unsupervised case. Based on that, we derive a new architecture to study the problematic of multi-domain image-to-image translation.

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

The idea was to understand the approach used by cycleGANs for Image-to-Image translation [2] and precisely in the unpaired setting (unsupervised), where the idea is to learn how to translate between domains without paired input-output examples. The first step was to fully understand the paper and the model architecture that consisted in learning two mappings from a source domain to a target domain and vice-versa, this is done by training two opposite GANs with convolutional neural networks as generators, and adding a consistancy cycle loss to make sure that the mapping between the two spaces distributions are coherents and consistant. And also we proceeded by evaluating our model results by using the entropy measure and inception score to evaluate the quality of our generated images, and which will later be evaluated on the proposed architecture (triCycleGAN).

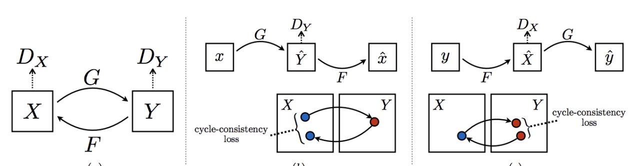

Figure 1: figure from [2] showing the mapping functions $G : X \to Y$ and $F : Y \to X$, and the associated adversarial discriminators D_Y and D_X in the cycleGAN setting.

1.1. Datasets

The datasets were collected from imageNet, we worked with a data set to reproduce the training process and get familiar with the code which is typically composed of pandas and polar bears images "panda2polarbear", reproducing the work done in the original paper with horse2zebra image translation model, and another dataset of three classes : apple, orange and lemon, that was used to produce results from 3 cycleGANs and the triCycleGAN model (cf: 3.TriCycleGAN). The data was resized and organized in folders by class and train/validate split in order to be adapted to the code used to implement [2], and also to test the TriCycleGAN model on it.

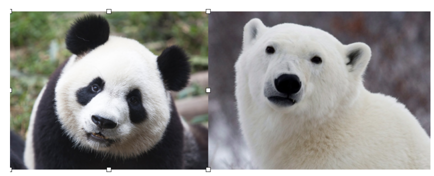

Figure 2: samples from panda2polarbear dataset collected from imageNet

Figure 3: samples from the orange-apple-lemondataset collected from imageNet

2. Multi-domain Image-to-Image Translation : TriCycleGAN

In this section, we present the architecture of TriCycleGAN which is heavily inspired by vanilla CycleGAN in order to perform multi-domain image-to-image translation.

2.1. Architecture

The architecture of TriCycleGAN consists of three generators and three discriminators creating a cycle between three domains X, Y and Z.

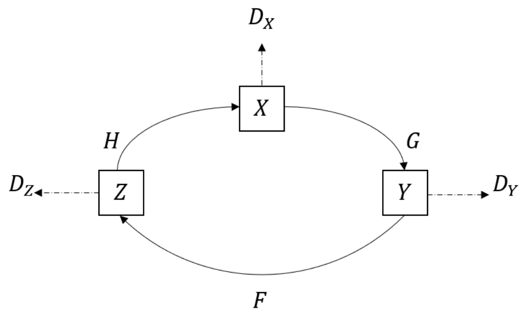

Figure 4: Architecture of TriCycleGAN

In this case, the inverse mapping of $G : X \to Y$ is the two mapping composition defined by $H \circ F : Y \to X$. The same principle applies for F and H.

2.2. Loss function

The loss function is also defined in a similar way by considering :

- each GAN's adversarial loss : $\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1 - D_Y(G(x)))] + \mathbb{E}_{z\sim p_{data}(z)}[\log(1 - D_Y(G \circ H(z)))]$
- a cycle-consistency loss : $\mathcal{L}_{cyc}(G, F, H) = \mathbb{E}_{x\sim p_{data}(x)}[||H \circ F \circ G(x) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G \circ H \circ F(y) - y||_1] + \mathbb{E}_{z\sim p_{data}(z)}[||F \circ G \circ H(z) - z||_1]$
- eventually an identity loss : $\mathcal{L}_{identity}(G, F, H) = \mathbb{E}_{x\sim p_{data}(x)}[||H(x) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G(y) - y||_1] + \mathbb{E}_{z\sim p_{data}(z)}[||F(z) - z||_1]$

An important point that needs to be highlighted concerning the different loss functions is the way they are weighted. Indeed, in the context of three domains with the same cardinality, each discriminator receives twice as many fake images than real images. Therefore, we multiply by 2 the terms corresponding to real images (for example $\mathbb{E}_{y\sim p_{data}(y)}[\log D_Y(y)]$, ...)

2.3. Advantages

Using TriCycleGAN instead of an equivalent architecture using CycleGANs has many advantages such as :

- The architecture is extendable to $n \geq 3$ domains
- The architecture achieves respectable performances compared to an equivalent CycleGAN solution trained with the same number of epochs.
- There are less parameters to train. In fact, if we denote N_G and N_D the number of parameters to train for the generator and discriminator and $N_{GAN} = N_G + N_D$, then the total number of parameters to train for:
  - TriCycleGAN :
    $N_{TriCycleGAN} = nN_{GAN}$
  - CycleGANs with a fully-connected architecture:
    $N^{(1)}_{CycleGAN} = n(n-1)N_{GAN}$
  - CycleGANs with a ring-based architecture:
    $N^{(2)}_{CycleGAN} = 2nN_{GAN}$
  - CycleGANs with a star-based architecture :
    $N^{(3)}_{CycleGAN} = 2(n-1)N_{GAN}$

3. Results

We first evaluate the results on our reproduced cycleGAN work on our dataset classes, for unpaired image-to-image translation. In this partical unsupervised setting we don't have ground truth input-output pairs of images available for evaluation, so beside the natural human evaluation of the quality of the generated images and their similarity to the target class, we needed a better way to evaluate it quantitatively.

3.1. Qualitative results

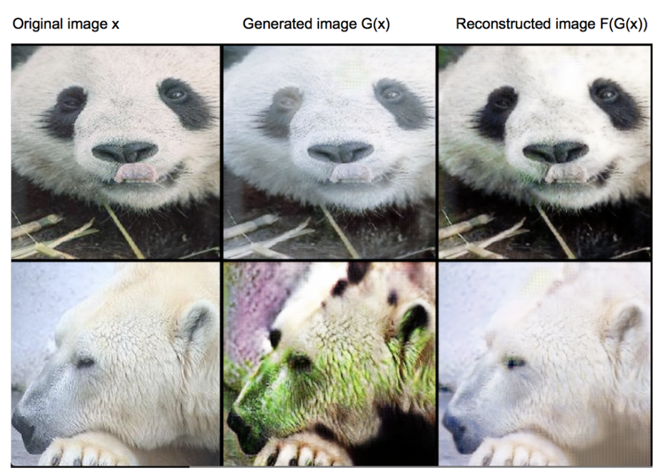

Figure 5: an example of image translation process (after 40 epochs training) on panda2polar bear dataset

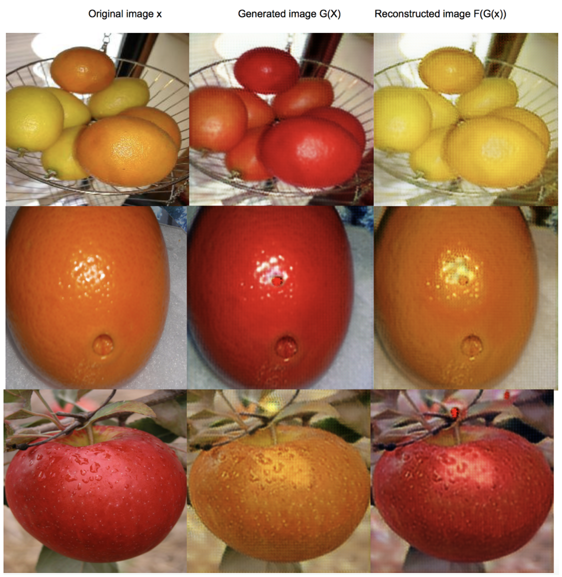

Figure 6: examples of image translation test of 3 cycle-GANs on orange-apple-lemon dataset

3.2. Quantitative results

We thought about a metric to evaluate our generated images that in someway is similar to the inception score [1]. We used a good pretrained classifier resnet18, that has the ability to give us a good approximation of the classes distribution (by simply using a softmax), and then using the entropy as a measure of the information loss.

The evaluation was made on the triCycleGAN results by comparing : Real classes images, fake images and reconstructed images.

In the quantitative evaluation part, we tried to evaluate to aspects : the first one is the target class image generation, in other words whether the generated image is really corresponding to the target class meant to be, and this was done by evaluating the pretrained classifier top output classes prediction to each pair (real-fake image) not necessarily the same image, but if the generation was well done the classifier should extract the final features in the image that should look quite similar and hence the final output classes should be similar.

In figure 7. for example the real lemon image (1) compared with the generated fake lemon (4) and (3) vs (2) :

Figure 7

Image 1	452	955	951
Image 4	107	452	883

Table 1: Top 3 classes prediction using a resnet18

In Table 1 are shown the mean entropy of each class on the generated (fake) images through the triCycleGAN sequences, while in Table 2 we evaluated the mean entropy on each class on the real original images.

	apples	oranges	lemons
Entropy	3.5	3.5	3.6

Table 2: mean entropy scores of the generated images of TriCycleGAN

	apples	oranges	lemons
Entropy	3	2.6	2.7

Table 3: mean entropy scores of real original images

We noticed through our evaluation, that the pretrained model doesn't always output the same class for both fake and real images but in most of the cases the top prediction class of the real images samples is always among the top 3 output classes in a fake image corresponding to the same class, as illustrated on an example in figure 8. Moreover, when we take a look at some of the results summarized in Table 1 and Table 2, we notice that for the first label (apple) the difference in the mean entropies values is not big, while for oranges and lemon data we notice a higher overall mean entropy compared to the real data, and that is on of the unstabilities that the triCycleGAN model may have sometimes when trained for longtime where one of the two classes (orange-lemon) tend to dominate.

4. Annex : other example

Figure 8: Real image

(a) Ukiyo-e

(b) Van Gogh

Figure 9: Obtained images with a TriCycleGAN

(a) Ukiyo-e

(b) Van Gogh

Figure 10: Obtained images with two CycleGANs

References

[1] S. Barratt and R. Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.