Final Computer Vision Project : SinGAN

Avigail Zerrad & Daniel Amar

Janvier 2020

1 Introduction

Introduced in 2014 [1], GENERATIVE ADVERSARIAL NETWORK consists of two models which influence one another :

- One known as generative model, which aims to find the probability distribution of input variables and generate similarly structured data.
- The other known as discriminative model, which aims to map a single possible output from the given input data. In other terms, it tries to figure out whether images are fake (generated thanks to the generator) or real.

The particularity of **GANs** is the use of a "min-max problem" with the combination of two adversarial CNNs where the generator tries to minimize its loss and maximize the loss of the discriminator (see figure 1).

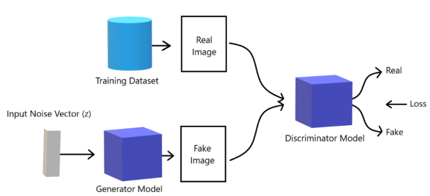


Figure 1: Structure of GANs

In this project, we will not focus on GANs but rather on an application known as **SinGAN** and developed in *SinGAN: Learning a Generative Model from a Single Natural Image* [2]. SinGANs only need one image and can serve for different purposes such as editing, harmonization, super-resolution or creating animations.

2 SinGAN

The main idea behind SinGANs is the use of a pyramid of convolutional GANs, where each is responsible for capturing the distribution of patches at a different scale (see figure 2). Thanks to this architecture, the model is able to learn global properties of the image as well as finer details. In practice, the generation of an image sample starts at the coarsest scale N. At this scale, the generation is purely generative, i.e the input for G_N is a white Gaussian noise z_N, and we have : $\tilde{x}_N = G_N(z_N)$.

If we denote x the original image, x_n is a downsampled version of x by a factor r^n, $(r > 1)$.

At each level $n > N$ of the pyramid, the generator G_n, whose input is the output of the previous scale (upsampled to the current resolution) plus a random noise image z_n, produces image samples according to the patch distribution in the corresponding image x_n. The associated discriminator D_n, identify patches in the generated samples $\tilde{x}_n$ from patches in x_n.

As we can see on the Figure 2, the effective patch size decreases as we go up the pyramid.

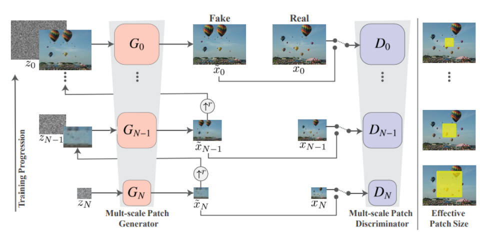


Figure 2: Architecture of SinGANs

3 Image Style Transfer

In order to compare the results obtained with SinGAN, we decided to use an other algorithm named Style Transfer [3]. The general idea of this algorithm is based on the observation that higher layers in a CNN capture the high-level content in terms of objects and their arrangement in the input image, and that the style can be reconstructed using different subsets of CNN layers. Then, the Neural Style Transfer method synthesizes the new image by jointly minimizing the distance from the content representation and the distance from the style representation.

Figure 3: Reproduced results from [2]

4 Paint-to-image

As we said above, SinGAN can be used in various image manipulation tasks and we decided for this project to focus on one application named Paint-to-image. The idea is to transform a paint into a realistic photo. In order to illustrate this notion, we first reproduced the results presented in the paper (figure 3). We notice that SinGAN gives more realistic results in comparison with Style Transfer but the image is not perfect since some cows have 6 legs.

5 Assistive product design

In this section, we decided to use SinGANs so as to make product design easier. The idea is that a designer may want to modify an existing product, making for instance its shape evolve. We decided to illustrate this idea by working with cars. We drew the sketches ourself with the *Paint* software. Our results can be summarized in the figure 4.

SinGAN The main observation we can do from these results is that sinGAN seems to work well on the global shape of the car, and also with some finer details.

- For the blue car, the rear light of the sketch slightly appears in the output. Car wheels are also well inspired from the training image. However, details like the door handle are too fine to be 'understood' by sinGAN. We can also observe that the colors of the output are well reproduced using the training image.
- For the black car, sinGAN produced a very interesting output ; we can see that it well 'understood' the new shape of the sketch and we clearly see the inspiration of the training image.

Figure 4: Results

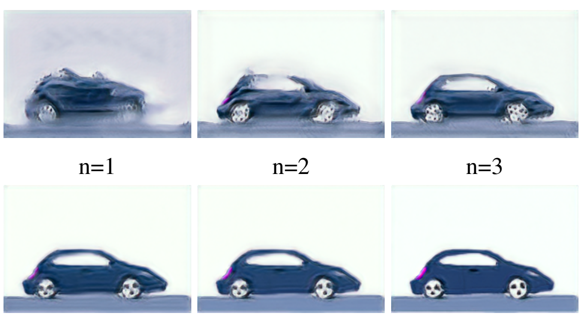
Neural Style Transfer

- For the blue car, we observe that in terms of color, Neural Style Transfer is quite bad. However, fine details such as headlights or door handle are much better than with SinGAN.
- For the black car, the result is not really convincing : the output is too close to the input.

From the obtained results, we can not say that one method is better than another. It seems that it depends on the images we work on. In the part of evaluation, we will confirm this idea.

Effect of the injection scale We illustrate with the following figure the effect of the injection scale on the results produced by sinGAN.

As we can see, if we inject the input too early in the pyramid, the output has not a good shape (n=1), whereas when we inject the input too late, the output is too close to the input (n=6).



n=1 n=2 n=3

(a) n=4 (b) n=5 (c) n=6

Figure 5: Effect of injection scale

6 Evaluation

6.1 Qualitative aspect

Evaluation for our problem is not simple and has often to rely on qualitative aspects. In our project, the 'best' algorithm is the one which gives the more realistic image, and which is a good combination of the trainig and the input. In order to evaluate the two algorithms, we realized a survey about which the respondent has to answer for each image (black and blue cars) this question : 'Which image is real?'. The question is willingly unclear and there are not other indications. We obtained the results (table 1) with a sample n=93.

	Style Transfer	SinGAN
Blue Car	33.3%	66.7 %
Black Car	52.5 %	47.5%

Table 1: Which image is real ?

6.2 Quantitative evaluation

We used the MSE and the SSIM metric [4] to quantitatively evaluate the results we obtained with both methods. Both metrics do not seem to be adequate to assess if the images are real since they mesure the similarity between the output and the training image which serves as a reference. SSIM gives a value between -1 and +1. The closer the SSIM value is to +1, the more similar the images are and conversely, the closer the SSIM value is to -1, the more different the images are.

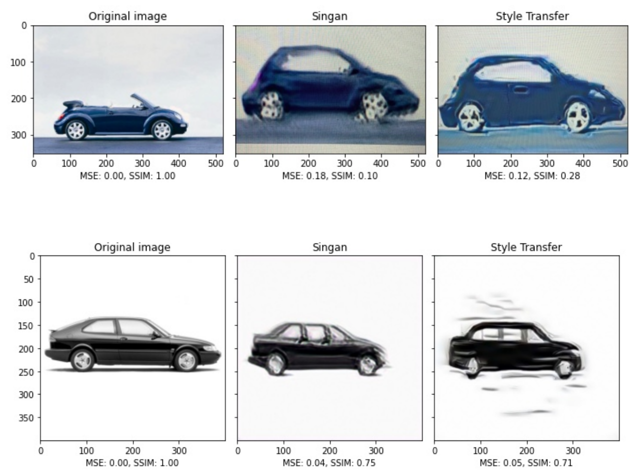


Figure 6: SSIM

7 Conclusion

To conclude, even if sinGAN produces interesting results using a single image, we can not say that this method outperforms the state-of-the-art. The great disadvantage of sinGAN is the important algorithmic cost (about 1h30 with a GPU) for a single image.

This project was very interesting and allowed us to experiment results of a research paper and to talk about its limits.

References

[1] Ian J. Goodfellow et all. Generative Adversarial Nets. (English). *Arxiv*, 2014. 1

[2] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Computer Vision (ICCV), IEEE International Conference on*, 2019. 1, 2

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity,. In *IEEE Transactions on Image Processing*, April 2004. 3

Annexe

Training Input Singan Style Transfer