INFO8010: Final project
CycleGAN for style transfer

Pierre Navez[1] and **Antoine Debor**[2]
[1] *pierre.navez@student.uliege.be (s154062)*
[2] *antoine.debor@student.uliege.be (s173215)*

I. INTRODUCTION

Given the sanitary context, more and more students are struggling to cope with the monotony of distance learning. Being in front of the screen all day, isolated from the campus and other students, seems to have a detrimental effect on motivation and, from a general point of view, on mental health. From then on, we thought that using deep learning to brighten up the current routine of students could be an interesting starting point for this project.

(a) Original (b) Styled (c) Cycled

(d) Original (e) Styled (f) Cycled

Figure 1: Transfer of post-impressionist style to the authors of this report, and cycle transformation (from the style transfer image to the original one). ℓ_1 difference between the original and the cycled: Antoine (top): 0.0831 and Pierre (bottom): 0.0985.

Our initial idea was to perform style transfer using cycleGANs [1] on lectures podcasts in order to make the speech of our teachers funnier. This style transfer would be done from the style of different great painters, as it has already been done for photos or film excerpts. Unfortunately, this project could not be realised before the deadline of the project and, due to the complexity underlying the training of cycleGANs, we focused on static images rather than videos. To extend the ability of our network to videos, a distillation phase would have been required to allow fast and in-real-time style transfers.
Our approach was originally to consider the general style of a given painter, including numerous and diverse paintings, rather than focusing on one of its masterpieces. Indeed, considering only one painting boils down the complexity of the task and would not exploit the whole capacity of cycleGANs (This kind of simple style transfer can for instance be easily achieved using *OpenCV* as presented here). In the following, this first trial will be referred to as the *Faces2VanGogh* (*F2V*) transfer. The paintings dataset used in this approach is the Best Artworks of All Time (BAAT) [2] dataset available on the *Kaggle* platform. The problems encountered with this original method will be discussed in section IV.
A second approach has been developed to avoid the previous one's issues. It consists in considering portraits paintings only, sampled from a pool of portraits classified according to which artistic movement they belong to. This dataset is the Portrait Painting Dataset For Different Movements (PPDDM) [3] gathered from Mendeley Data. As the *Faces2VanGogh* has been tested using Van Gogh paintings, only post-impressionism portraits have been used during the training phase of this second approach, in order to keep the same painting style. In the following, this second trial will be referred to as the *Faces2Impressionism* (*F2I*) transfer.
Before diving into proper painting style transfer though, we first tried to reproduce the results of cycleGANs' original paper [1] to validate our models and training methods. In particular, we used the Horse2Zebra dataset available on Kaggle and gathered from UC Berkeley's official directory of CycleGAN Datasets[1]. In the following, this reproduction trial will be refered to as the *Horses2Zebras* (*H2Z*) transfer.

II. RELATED WORK

The original CycleGAN idea has been developed in the "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" paper from Zhu et al. [1]. This paper comes with a vast collection of qualitative results and an easy-to-read and very helpful source code. Further developments have been made upon this original CycleGAN in the "Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data" paper from Almahairi et al. [4].
Style transfer on videos has already been performed in the original CycleGAN paper, and with other techniques in several papers among which the "Painting Style Transfer for Head Portraits using Convolutional Neural Networks" paper from Selim et al. [5], and the "Artistic style transfer for videos." conference paper from Ruder et al.[6].
Several implementations of the CycleGAN network can be found performing different kinds of style transfer. A

link to some of them is given in section VI.
This works thus combines a well-studied process, *i.e.* artistic style transfer, with cycleGAN; please note that this work is not the first one to attempt this.

III. METHODS

A. General principle

CycleGANs are generative adversarial networks (GANs) [7] performing image-to-image translation without the need of paired data. To review cycleGANs' principle in the scope of this work, let us consider that one has a set of post-impressionist paintings (corresponding to the X domain) at its disposal and, in addition, a set of real faces pictures (corresponding to the Y domain). The principle of a *CycleGAN* network is to have one generator-discriminator pair to apply post-impressionist style on the real pictures, along with another generator-discriminator pair which would perform the inverse transformation. A general schematic is presented in fig. 2.

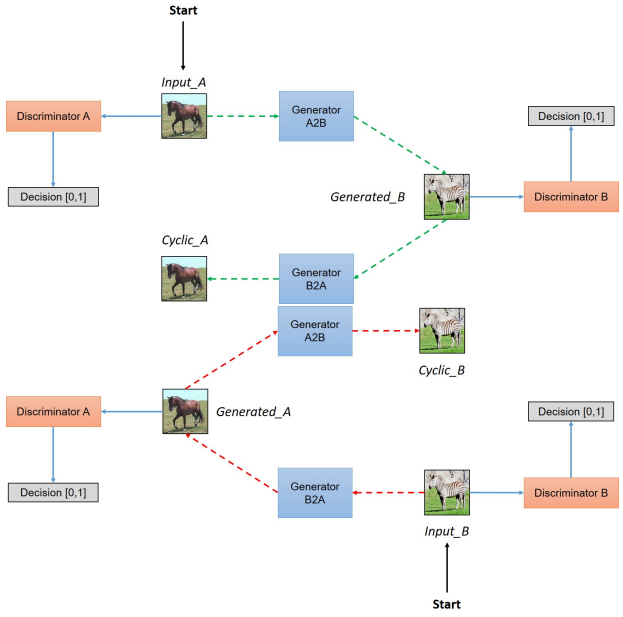


Figure 2: Simplified view of CycleGAN architecture ([8])

Formally, the goal is to learn a mapping function between the two domains X and Y given training samples $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and $\{y_j\}_{j=1}^{M}$ where $y_j \in Y$. The data distributions are denoted $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. The model includes two mappings $G : X \to Y$ and $F : Y \to X$. Additionally, there are two discriminators D_X and D_Y where D_X aims to distinguish images x from translated images $F(y)$ and D_Y aims to distinguish images y from translated images $G(x)$.

B. Losses definition

The objective contains three types of error to minimize: the *adversarial losses*, the *cycle consistency losses* and the *identity mapping loss*.

The *adversarial loss* enforces the matching between the distribution of the generated images and the target domain. For the mapping function $G : X \to Y$ along with the discriminator D_Y the objective is expressed as:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = E_{y\sim p_{data}(y)}[\log D_Y(y)] + E_{x\sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \quad (1)$$

In the context of a zero-sum game between G and D_Y, the generator tries to minimize the objective that the discriminator tries to maximize. The same loss exists for the mapping function $F : Y \to X$ along with D_X. In practice, this loss is implemented via the means squared error; further information can be found in the source code, in particular `training.py`.

The *cycle consistency loss* is introduced to ensure that the mapping derived from the first objective not only corresponds to a similar data distribution from one domain to another, but is also corresponding visually. Indeed, with networks of large enough capacity, a mapping derived from the adversarial loss can correspond to the data distribution of the target domain but not provide a good *qualitative* result. The idea here is to say that the mapping should be cycle consistent. If we apply a transformation from image x from domain X to image y in domain Y, then transforming back the image y should provide the image x as well. Formally, $x \to G(x) \to F(G(x)) \approx x$ and the same for y and $G(F(y))$. So the cycle consistency loss is expressed as

$$\mathcal{L}_{cycle}(G, F) = E_{x\sim p_{data}(x)}[||F(G(x)) - x||_1] + E_{y\sim p_{data}(y)}[||G(F(y)) - y||_1]. \quad (2)$$

For a broad variety of problems, the overall objective function can thus be expressed as the sum of the objectives with a parameter λ_{cycle}, encoding the importance of the cycle consistency besides the adversarial losses

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda_{cycle}\mathcal{L}_{cycle}(G, F). \quad (3)$$

However, according to the original paper [1], it is helpful to introduce an additional loss when considering photo generation from paintings. This third loss encourages the mapping to preserve color composition between the input and output. The generator is thus regularized according to Taigman et al.[9], *i.e.* by implementing the following *identity mapping loss*:

$$\mathcal{L}_{identity}(G, F) = \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(y) - y\|_1\right] + \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(x) - x\|_1\right]. \quad (4)$$

This third loss can be added to the overall loss defined by eq. (3), multiplied by a coefficient $\lambda_{identity}$. As this loss is similar to $\mathcal{L}_{cycle}$, note that it is also multiplied by λ_{cycle}.

An important point has to be mentioned here: no normalization is used for the losses using the ℓ_1 norm. Indeed, it has been observed that, although providing reduced losses, adding a normalization factor (equal to $number\ of\ channels \times height_{image} \times width_{image}$) to scale the losses leads to very poor results. Abandoning this normalization strategy has been a real turning point in the improvement process. It is noteworthy that such a normalization is not present in the CycleGAN original paper's implementation, though.

C. Generators and discriminators architecture

Both generators and discriminators are convolutional neural networks, as we process images.
The two generators are made of three main parts: an encoder, a transformer and a decoder. The encoder consists in an initial layer followed by two downsampling ones. The transformer consists in several basic convolutional *ResNets* blocks. The number of *ResNets* depends on the size of the input image: there are 6 blocks if the size is equal to 128×128 or below, and 9 blocks if the size is equal to 256×256 or above. The decoder is made of two upsampling layers followed by a final layer. All convolutional layers use a combination of *Instance Normalization* [10] and, except the last one, a *ReLU* as activation function. The *ResNets* however use an identity activation function at the output.
The two discriminators each implement a *PatchGAN* [12]. This kind of discriminator evaluates structure at the scale of local image patches and tries to classify each patch in an image independently, hence acting as a texture discriminator.
The described architectures can be visualized in the schematic of fig. 3. Note that, according to the original paper's implementation, it is advised not to use the *Sigmoid* activation function in the discriminator's last layer when implementing *least-squares GANs* (*LSGANs*), as it is the case in the scope of this project (*cfr.* the adversarial loss' implementation).

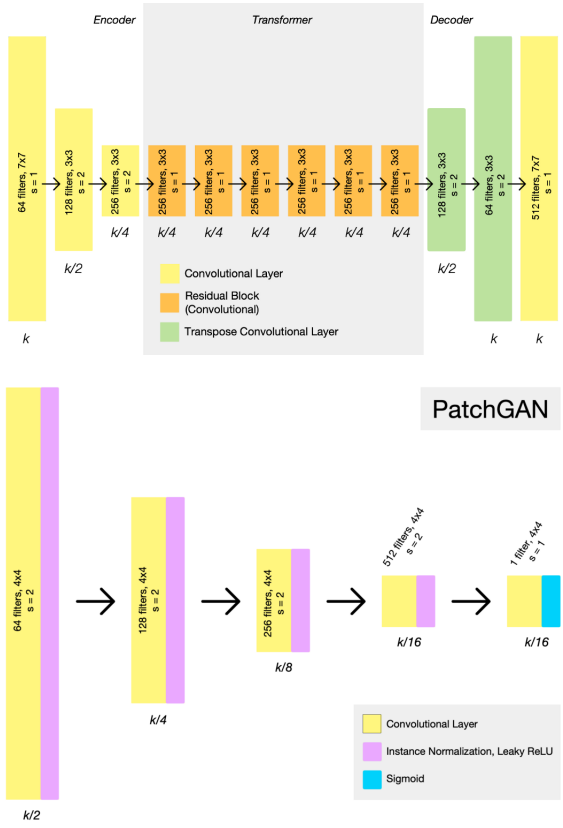


Figure 3: Generator (above) and discriminator(below) architectures ([11], modified)

For information, table I summaries the usage of the different datasets in the scope of this project.

Dataset	Total size (in PNG files)	Used size (in PNG files)
FFHQ	70k	1.1k
Horse2Zebra	2261	2261
BAAT	16.8k	870
PPDDM	927	240

Table I: Datasets specifications

D. Data

In addition to the datasets mentioned in section I, the human faces dataset used in the scope of this project is the Flickr-Faces-HQ Dataset (FFHQ) from NVIDIA Labs. According to the dedicated repository, this dataset, which gathers its content from *Flickr*, consists of high-quality PNG images and contains considerable variation in terms of age, ethnicity and image background. It also has good coverage of accessories such as eyeglasses, sunglasses, hats, etc.

E. Training process

The different hyperparameters used in the different training processes are presented in table II, where X and Y correspond to the $X2Y$ process. In this table, "a (virtually b)" means that the corresponding *PyTorch* dataset is built such that it is considered as having a size equal to b during the training process, while containing only a images though. This is achieved by modifying the `__len__` method of the *PyTorch* `Dataset` class. Several additional remarks are noteworthy:

- The learning rate has been taken equal for the generators and discriminators and kept constant during the whole training phase for each of these, although the original paper implements a decaying learning rate. This is an arbitrary decision and the decaying variant has not been tested, due to a lack of time. However, this could be a refinement of the work presented here as scheduling often provides improvements in terms of results.
- Similarly as in the original CycleGAN paper, the optimizer is the Adam [13] optimizer implemented in *PyTorch*.
- The size of the images is the same for all approaches and is equal to 128×128.
- In the $F2I$ approach, *impressionist* portraits are used in the test phase, rather than *post-impressionist* portraits. Indeed, these two artistic movements show similar patterns and, since the post-impressionist dataset is quite small, using a smaller fraction of it would have yielded very poor results. Since this test set is involved in $painting \rightarrow faces$ transfers (*i.e.* applying a "human face" style to a painting), the cycleGAN should provide similar results as the ones which would have been observed with post-impressionist test images.

	H2Z	*F2V*	*F2I*
# of epochs	100	100	100
X trainset size	1067	1000	1000
Y trainset size	1067 (out of 1334)	800 (virtually 1000)	170×4 (virtually 1000)
X testset size	120	100	100
Y testset size	140	70	70
Batch size	10	10	10
Learning rate	2e-4	2e-4	2e-4
Adam's β's	(0.5, 0.999)	(0.5, 0.999)	(0.5, 0.999)
λ_{cycle}	10	10	10
$\lambda_{identity}$	0.5	0.5	0.5

Table II: Hyperparameters for the different training phases

To enhance the robustness of the network and to increase the number of images at disposal, we performed data augmentation on the portraits dataset, which is quite small. Three transformations (implemented in *PyTorch*) have been applied separately to each image in addition to a resizing and a normalization: an horizontal flip of the image, a random affine transformation and a random rotation of the image between $-20°$ and $+20°$. These three transformations are responsible for the "$\times 4$" in table II. We also tried to introduce some Gaussian noise in the images and to perform vertical flips, but these yielded poor results and were judged useless in the scope of this project.

IV. RESULTS

In this section, we present our results for the three different style transfer trials mentioned above. For each of these, we first present qualitative results, *i.e.* generated images besides original ones, followed by quantitative results, presenting the evolution of the different losses.

For the qualitative results, we show both $A \rightarrow B$ and $B \rightarrow A$ results for a given $A2B$ transfer. Indeed, for all trials, no difference in terms of importance has been introduced in the training process regarding a preferred direction of transfer. We first show images generated during the training phase. Then, we show a selection of images generated using the trained generators on new images. As the amount of training images presented in this work is already large, we decided to show only four images per transfer for the testing phase (in addition to those presented in fig. 1). Please note that, for the sake of simplicity, only a small fraction of the test sets described in table II has been used. In each figure, images are grouped by four, in a fixed order. They correspond respectively to the **original** image (*i.e.* x or y), the **identity-mapping** image (*i.e.* $F(x)$ or $G(y)$), the **transferred/translated** image (*i.e.* $G(x)$ or $F(y)$) and the **cycle** image (*i.e.* $F(G(x))$ or $G(F(y))$).

Note that, in the scope of this project, the quantitative part does not bring much information, and the qualitative part is the most relevant. Several training losses are used to train the whole model, as explained in section III, but it has been decided to present only the two combined ones, *i.e.* the global generator and discriminator losses, rather than each individual adversarial loss, each identity-mapping loss, etc. Furthermore, no testing curve have been generated since the test data has been used after the training to assess the qualitative performance of our models. However, to still provide quantitative test results, the ℓ_1 difference between the original image and the cycle one is given for each image.

A. *Horses2Zebras*

The results presented in the CycleGAN original paper present various style transfers. The most famous is without a doubt the *Horses to Zebras* style transfer, in which they manage to generate a zebra from a horse image. Black and white stripes appear on the skin of the horse, and only on the skin. Indeed, the network manages to extract the semantic data of the horse image to identify the horse, in order to apply the "zebrification" on its skin and not on the background, which is impressive.

In order to validate our implementation, we tried to reproduce these results, or at least to reach a certain level of quality regarding the aim of this style transfer.

1. *Qualitative results*

Qualitative results obtained during training can be seen in fig. 4 for the horse → zebra transfer and in fig. 5 for the zebra → horse transfer. Qualitative results obtained during testing can be seen in fig. 6 for the horse → zebra transfer and in fig. 7 for the zebra → horse transfer.

Figure 4: Training results for *H2Z* (H→Z) (100^{th} epoch)

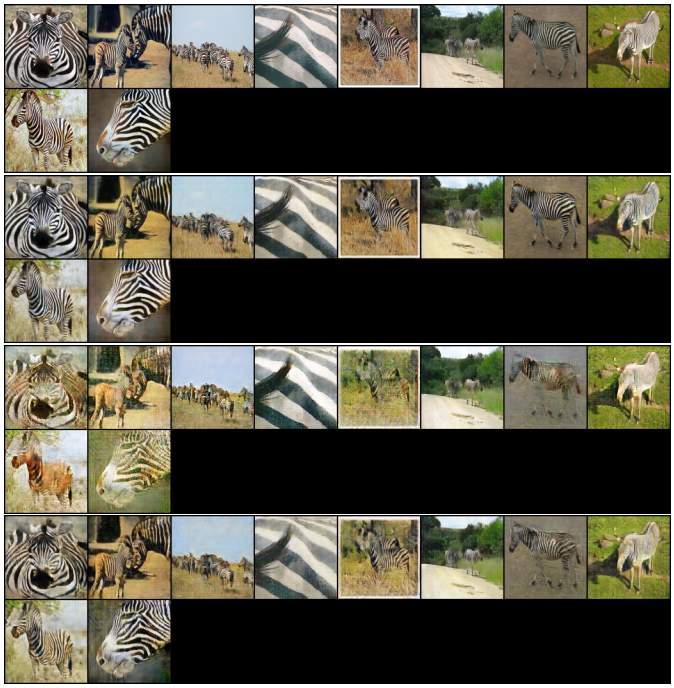


Figure 5: Training results for *H2Z* (Z→H) (100^{th} epoch)

Figure 6: Testing results for *H2Z* (H→Z)

2. *Quantitative results*

The two above-mentioned training losses for *H2Z* are presented in fig. 8a and fig. 8b.
The ℓ_1 difference between the original images and the cycle ones for *H2Z* are presented in table III. The left to right progression of fig. 6 and fig. 7 corresponds to the top to bottom progression in table III.

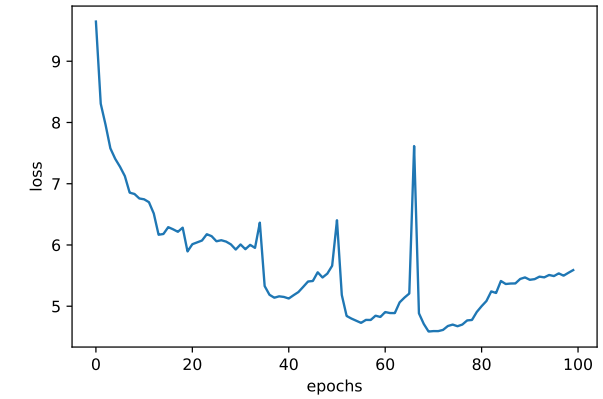
Image	H → Z transfer	Z → H transfer
extr. left	0.0919	0.2751
left	0.0792	0.1120
right	0.1163	0.1368
extr. right	0.1423	0.1906

Table III: ℓ_1 cycle differences for *H2Z*

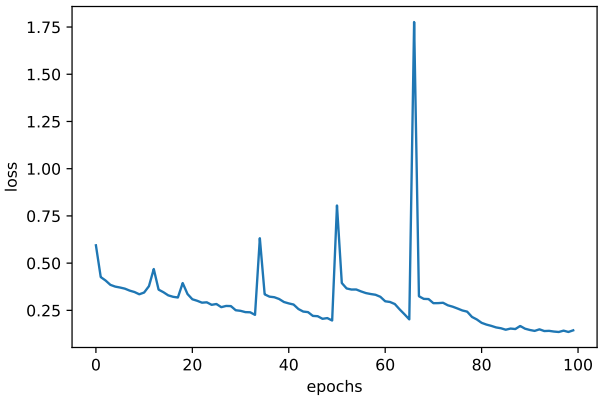
B. *Faces2VanGogh*

As mentioned in section II, artistic style transfer is a frequent subject of interest in generative networks. Various papers, blogs or notebooks present a wide range

Figure 7: Testing results for *H2Z* (Z→H)



(a) Generator loss



(b) Discriminator loss

Figure 8: Training losses for *H2Z*

of such transfers. In the scope of this project, the goal is to perform such a transfer using CycleGAN, from real human faces to paintings. In the Best Artworks of All Time dataset, the largest painter represented is Vincent Van Gogh. We therefore decided to use its paintings to perform the desired transfer. However, this naive approach does not take into account the fact that, when using CycleGANs, both the original images and the target ones should present similar semantic data. In this case, although some Van Gogh paintings are portraits, a vast majority represent landscapes or indoor scenes, such as the famous *Van Gogh's chair*. The expected results of such an approach are thus not trivial to define. Indeed, transferring the style of a painting of a chair to a human face can be conceptualized, but the opposite is more tricky to consider. Furthermore, as stated in [11], transfers implying substantial geometric changes to the image usually fail.

1. *Qualitative results*

Qualitative results obtained during training can be seen in fig. 9 for the painting → face transfer and in fig. 10 for the face → painting transfer. Qualitative results obtained during testing can be seen in fig. 11 for the painting → face transfer and in fig. 12 for the face → painting transfer.

2. *Quantitative results*

The two above-mentioned training losses for *F2V* are presented in fig. 13a and fig. 13b.
The ℓ_1 difference between the original images and the cycle ones for *F2V* are presented in table IV. The left to right progression of fig. 11 and fig. 12 corresponds to the top to bottom progression in table IV.

Image	V → F transfer	F → V transfer
extr. left	0.2955	0.1382
left	0.1928	0.1522
right	0.1653	0.2073
extr. right	0.2340	0.1830

Table IV: ℓ_1 cycle differences for *F2V*

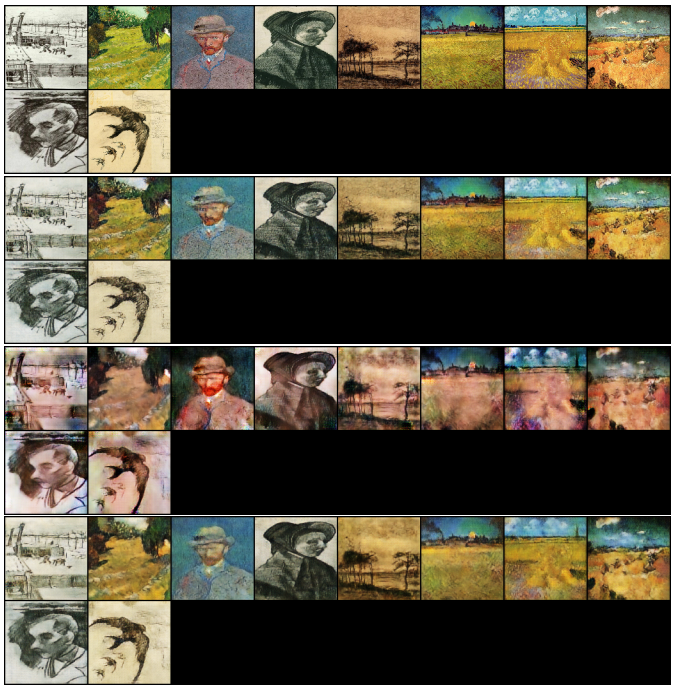


Figure 9: Training results for *F2V* (V→F) (80^{th} epoch)

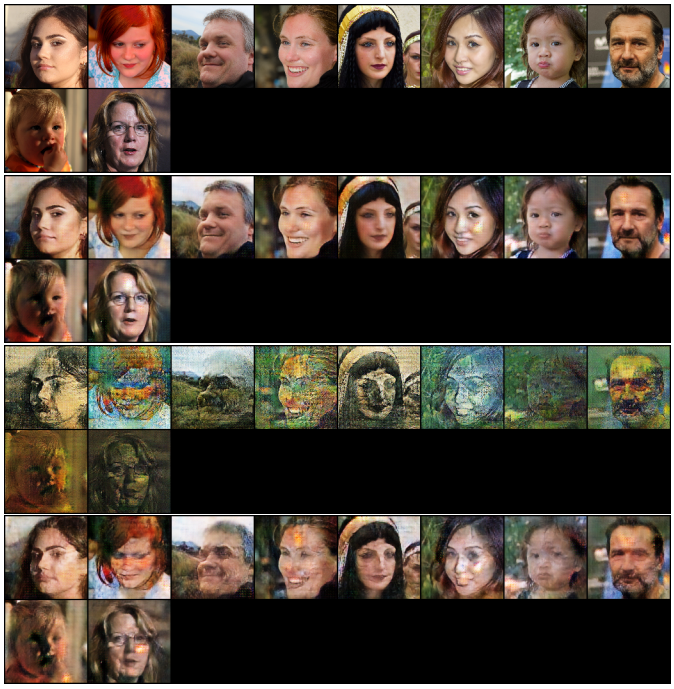


Figure 10: Training results for *F2V* (F→V) (80^{th} epoch)

C. *Faces2Impressionism*

To avoid the issue of semantic content mentioned in the *Faces2VanGogh* results section, two approaches have been considered. The first one consists in considering paintings of portraits only, in order to bring the semantic contents of the two datasets closer to each other. This approach obviously makes sense only if the portraits come from a same artistic movement, in order to guarantee a consistency in the generated data. This approach has been tested using the above mentioned Portrait Painting Dataset For Different Movements's post-impressionism folder and the results are presented from fig. 14 to fig. 17. A second approach would have been to generate a new painting dataset using a simpler tool as the one mentioned in section I which makes use of *OpenCV*. With such a network, it would have been possible to generate stylized portraits from the faces dataset and using post-impressionism paintings as the style to transfer source. This approach has not been implemented in the scope of this project. However, this could be an interesting and refining approach to test as this would likely bring the semantic contents of the datasets even closer to each other than with the first approach.

Figure 11: Testing results for *F2V* (V→F)

1. *Qualitative results*

Qualitative results obtained during training can be seen in fig. 14 for the painting → face transfer and in fig. 15 for the face → painting transfer. Qualitative results obtained during testing can be seen in fig. 16 for the painting → face transfer and in fig. 17 for the face → painting transfer.

Figure 12: Testing results for *F2V* (F→V)

2. *Quantitative results*

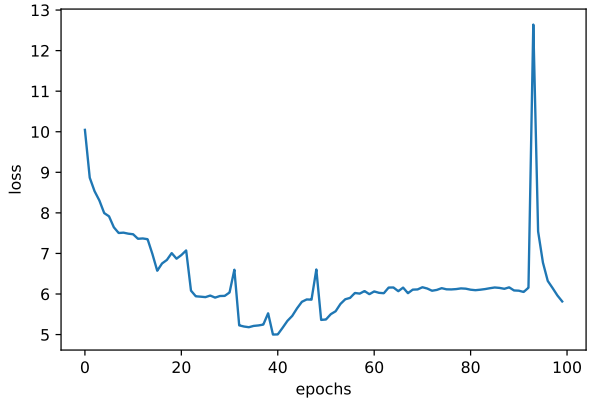
The two above-mentioned training losses for *F2I* are presented in fig. 18a and fig. 18b.
The ℓ_1 difference between the original images and the cycle ones for *F2I* are presented in table V. The left to right progression of fig. 16 and fig. 17 corresponds to the top to bottom progression in table V.

Image	I → F transfer	F → I transfer
extr. left	0.1370	0.0867
left	0.1130	0.1432
right	0.1297	0.1537
extr. right	0.1253	0.1172

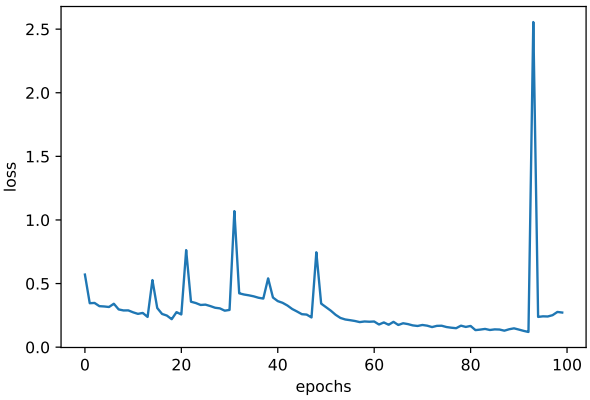
Table V: ℓ_1 cycle differences for *F2I*

V. DISCUSSION

In this section, the various results presented in section IV are discussed. A global remark can be stated though: although the qualitative results may be better, the achieved style transfers show interesting properties. Naturally, the implementation, the parameters and the choice of the data could be refined in order to provide



(a) Generator loss



(b) Discriminator loss

Figure 13: Training losses for *F2V*

transfers of better quality, but some results, in particular for *H2Z*, are quite satisfactory. There are also numerous techniques in the literature to enhance the quality of generated images. However, in the scope of this project, none of them has been implemented nor tested. Also, as previously mentioned, some results were expected to be quite bad, in particular the painting to face transfer from *F2V*. However, some interesting patterns can still be noticed and discussed.

In the following, only testing data will be discussed. As stated in [11], translations on the training data often look substantially better than those done on test data, and are thus not as relevant as the testing ones to assess the performance of the model.

The results of *H2V* are far from being as good as in the original paper. However, it can be seen that the majority of the presented horses are quite well 'zebrified'. Naturally, it is clear from fig. 6 that the black horse is not well handled. It has been observed that both all-black

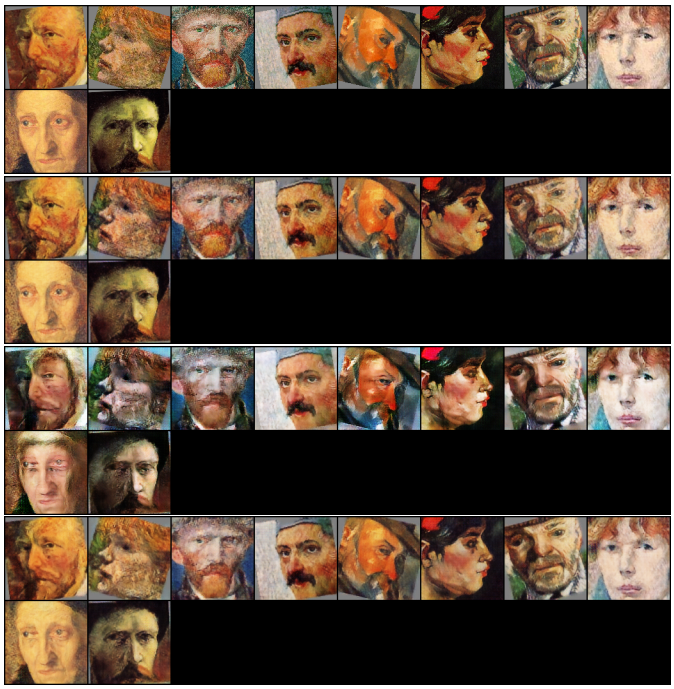


Figure 14: Training results for *F2I* (I→F) (80^{th} epoch)

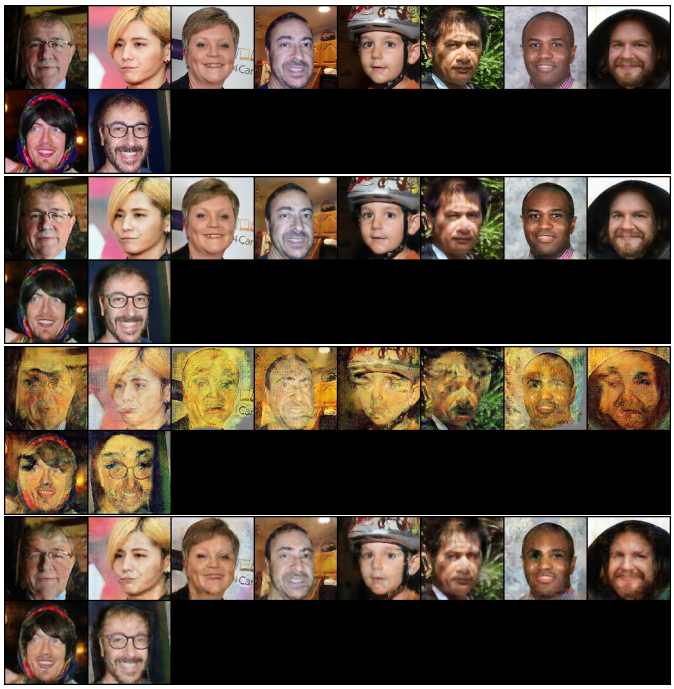


Figure 15: Training results for *F2I* (F→I) (80^{th} epoch)

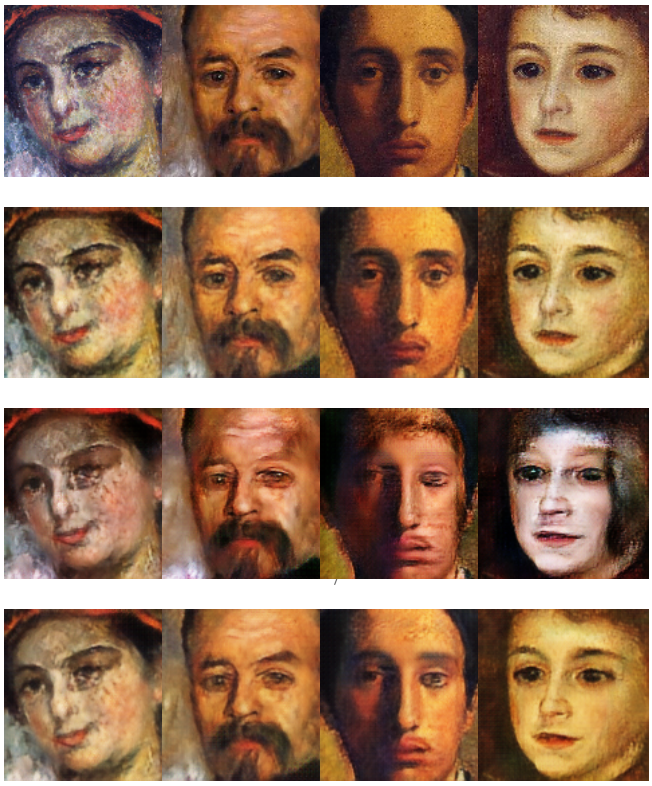


Figure 16: Testing results for *F2I* (I→F)

and all-white horses cause trouble to the model. Indeed, our cycleGAN is most of the time not able to properly add stripes to these types of images. The cause of this issue might be the "saturated" character of such colors, causing the model to misidentify the semantic content of the data. A similar effect has been observed for pictures in which the animal is far from the observer. In such cases, the model is not able to identify the animal and to translate it as it is too small.

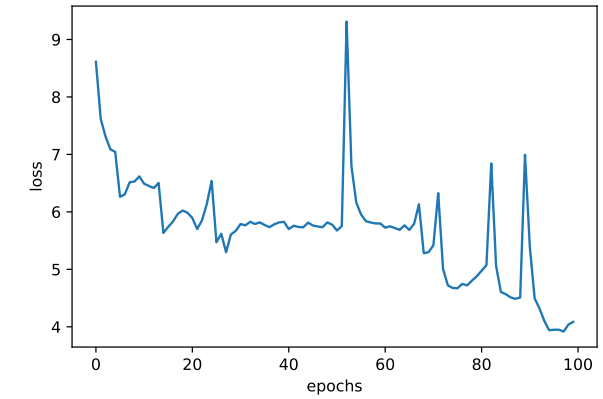
Another issue of the horse to zebra transfer is the loss of resolution, which makes the generated images appear blurry. This issue is a general one and applies to the two other studied transfers too. Again, several techniques in the literature aim at enhancing this resolution, but none has been considered in the scope of this project.

Despite this resolution issue, the identity mapping as well as the cycle transformation show pretty good results in terms of reconstruction. Indeed, the horses are always well re-generated and translating the generated zebra back to the original horse is also well achieved. This interesting property is a general one and applies to the two other studied transfers too. Obviously, the cycle image is always of lower quality regarding the original one, as it has to be reconstructed from a translated one.

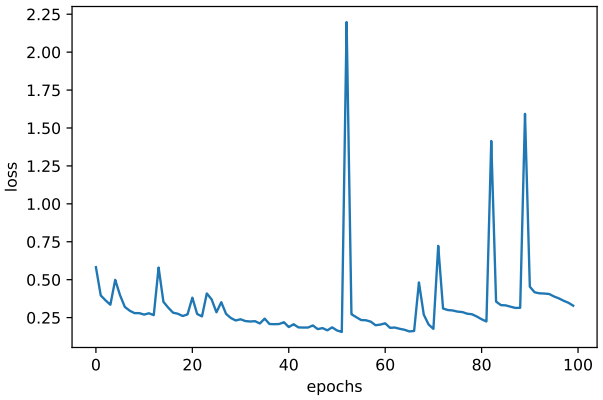
It is quite obvious that the zebra to horse transfer does not work as well as the opposite though. Indeed, even if brown patches are sometimes applied on zebras, the transfer is not satisfactory to the human eye. Moreover, the model sometimes "erases" parts of the animals, as it can be seen in the training images. These issues may have the same origin as the "saturation" one previously mentioned. Indeed, a zebra being made of all-black and all-white stripes, these issues are likely to be related.

Concerning the learning curves, not much can be said. Indeed, the global shape of these curves show that

Figure 17: Testing results for *F2I* (F→I)



(a) Generator loss



(b) Discriminator loss

Figure 18: Training losses for *F2V*

both generators and discriminators seem to learn, while presenting some learning instability. This is not surprising when considering adversarial training, but could as well come from an implementation error. Note that this instability corresponds to the peaks present in the curves, which seem correlated between the generators and the discriminators. This behaviour of the curves is similar for each experiment and will not be analyzed again in the remaining. An important remark has however to be noted: in early implementations, a convergence issue occurred because of the too good discriminator's ability to discriminate generator's creations at the beginning of the training. This caused the generator's training curve to monotonically increase and provided poor results.

As expected, the results of $F2V$ are not really impressive. The identity-mapping still show good results but the cycle one is not as good as in $H2Z$.
For the painting to face transfer, no clear behaviour could be expected. Indeed, it is not easy to determine what a painting of a pair of shoes should look like once translated with a human-face style. However, for paintings of portraits, it can be seen that the model still shows interesting results, as the skin becomes more human-like, as well as the hair. For paintings of landscapes/streets, it can be seen that the model tries to introduce human-like patterns though.
For face to painting transfers, the cycle transformation show quite poor results in some cases. The style transfer is not satisfactory neither, as some strange and not wanted patterns are applied to images. This is mainly due to the poor choice of the painting dataset. Indeed, as the content of these paintings varies a lot inside the dataset, the model is not able to learn a proper way of proceeding. This issue is however not really represented in the learning curves, as they are similar to those obtained for $H2Z$.

As previously stated, the aim of $F2I$ was to avoid the issues of $F2V$ *via* a better choice of dataset. It is actually quite difficult to assess the performance of this model though, as the transfers are not optimal. However, it can be noticed that the painting to face transfers are of much better quality than for $F2V$. It can be seen that the portraits are sometimes quite well modified in order to look 'human-like'. Still, some artifacts remain, and the model is far from perfect, but the eyes and the lips are in particular quite interestingly modified, as well as the skin.

As it can be seen in fig. 17, the face to painting transfers are of various quality and some generated images could be considered as quite satisfactory, while other ones could not. The transfers applied on the authors of this work presented in fig. 1 are for instance not that bad, and present an interesting style. The obtained results are however globally not really impressive and one could expect generated images closer to real paintings than what has been observed. This work is thus open to further improvements and refinements.

Concerning the ℓ_1 difference between the original images and the cycle ones, the amount of generated data is too small to draw any rigorous conclusion. However, one can observe that, for $H2Z$, the H→Z→H cycle seems to provide better results than the Z→H→Z cycle. For $F2V$, the F→V→F cycle seems to provide better results than the V→F→V cycle, except for the unique portrait painting, which is not surprising. For $F2I$, no direction of transfer seems to provide better results than the other. From a general point of view, $F2I$ seems to provide better reconstructions than the two others, while $F2V$ seems to provide the worst results. Again, no rigorous conclusion can be drawn though.

VI. USEFUL LINKS

Our Python implementation using Pytorch can be found in the submitted archive. The original paper and several implementations can be found on the CycleGAN Project Page. Additional inspiring work can be found here and here.

VII. ADDITIONAL TESTS

To validate the implemented architecture, a simple GAN has been built using the generator and discriminator detailed in the above. This network has been trained to overfit a single image of the portrait paintings. It however showed results that were not very satisfactory as the CycleGAN generator and discriminator architectures are quite different from the simple GAN's ones.

A famous regularization technique when it comes to GAN convergence is the R_1 regularization [14], adding the regularization term of eq. (5) to the discriminators' loss.

$$R_1(\phi) = \frac{\gamma}{2}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\|\nabla_{\mathbf{x}} d(\mathbf{x};\phi)\|^2\right] \tag{5}$$

In eq. (5), $\gamma > 0$, d stands for the discriminator, ϕ represents the parameterization of d and $\mathbf{x}$ represents the object for d to discriminate. It has been introduced to improve our CycleGAN's performance, but it did not show any relevance in the scope of this project. We thus decided not to keep it in the implementation.

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[2] Icaro. Best artworks of all time, 2019.

[3] Jiaqi Yang. Portrait painting dataset for different movements, 2021.

[4] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data, 2018.

[5] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, pages 129:1–129:18, 2016.

[6] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. *Rosenhahn B., Andres B. (eds) Pattern Recognition. GCPR 2016. Lecture Notes in Computer Science, vol 9796. Springer, Cham.*, 2016.

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

[8] Hardik Bansal and Archit Rathore. Understanding and implementing cyclegan in tensorflow.

[9] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017.

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[11] Sarah Wolf. Cyclegan: Learning to translate images (without paired training data), 2018.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[14] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018.