Deep Generative Modeling:
Neural Networks for Creativity Emulation
Experiments on Monet paintings benchmark

Corina Dimitriu

Abstract

This paper investigates at least two generative frameworks in detail for synthesizing pictorial art, taking Monet's paintings as a baseline for evaluating their performance in terms of generative quality, in the first place, but also time, stability and scalability, in the second place. The first framework is provided as a blending of many influential research papers, having their parameters — including core models — adapted for the previously stated experimental configuration, whereas the second one represents a personal contribution in the field of unpaired image-to-image generative tasks, combining generative adversarial networks and Stable Diffusion models into an expressive and highly interactive pipeline. As learning the structure and style details is a matter of extracting and integrating spread out dependencies, the experimental results are based on employing the U-Net architecture and eventually consist of competitive FID scores, proving to be a promising landmark for future work and reliability on both frameworks. It is important to mark the contextual authenticity of the synthesized pieces, all the more because of a quite reduced dataset, which generalizes to extending the collection of the works of art which many artists created during a lifespan.

Keywords: GAN, Stable Diffusion, generator, discriminator, U-Net, painting, unpaired.

Introduction

On the premise of producing pictorial art with high degree of fidelity and connectivity, this paper elaborates generative architectures on the edge of a Kaggle competition topic, which requires evolving Monet-style artifacts, starting from an unpaired collection consisting of paintings and a suite of photographs, mostly capturing natural landscapes. The approached Kaggle competition is called *"I'm something of a Painter Myself"* and suggests the use of adversarial generative networks for solving the artificially generating art problem, in particular for mimicking the color and brush Claude Monet would have used [1]. As it can be speculated from the dataset structure, this requirement can be translated into a style transfer problem, which applies the style characteristics it infers from the paintings to the base construction the photos depict. On the other side, as Stable Diffusion is the most renowned state-of-the art, currently debated model for generative tasks, another solution would consist in making use of the outline it follows. However, the contribution this paper invests in combines the generative adversarial technique, as it is demanded to be compulsorily integrated in the solution by the competition's statement, and the Stable Diffusion outline. As a metric for results evaluation, the MiFID (Memorization-informed Fréchet Inception Distance) score — an extension of the classical FID score — is considered appropriate by the competition statement, since it balances realism and diversity [2]. Therefore, the first solution implemented eventually resulted in a competition score of 39.87871 which allowed the undersigned to be on the 7**th** place. As there will be made clear in Chapter III, each of the two frameworks is evolved through a series of attempts regarding the models and losses employed by the generator and the discriminator, as well as the parameters, thus multiple variants of each solution will be presented.

The first solution builds a CycleGAN model with a suite of improvements regarding the loss functions and parameters. It is important to note that CycleGANs, with various improvements, are still considered a viable baseline for state-of-the-art approaches in the field of unapired image-to-image style transfer and that, due to the fact that the complex parameters setup requires many

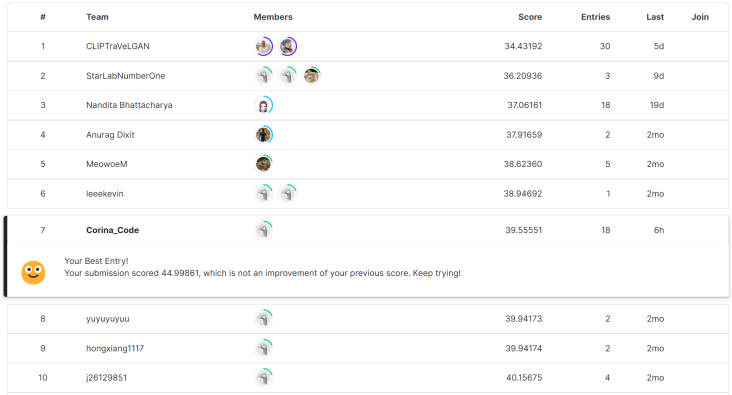


Figure 1: Screenshot capturing the Kaggle leaderboard of the competition.

modifications until stricking the right balance between generation and discrimination and between the consistuent parts integrated by each of them, this model consumes lots of computational resources and human effort.

In addition to the inability to fully explain the well functioning of the first solution, declared by the majority of related papers on the topic, the second solution intends to tackle the problem while also relying on heavy mathematical background. This solution replaces both the generator's and the discriminator's classical structure with a brand new, conclusive Stable Diffusion based architecture. To summarize the model which will further be thoroughly presented within Chapter III, the generator is represented by a Stable Diffusion model and the discriminator is characterised by a double U-Net architecture with CutMix augmentation and patch segregation at the loss level. One key difference with respect to the classical GANs is the missing forward call to the diffusion model itself within the generation performed at the training step, for the purpose of higher time efficieny and lower memory consumption. This call is substituted by a call to the double U-Net model employed by the Stable Diffusion generator, due to the noise–photo–painting chain it strives to output, since the paintings generator is assumed to firstly sample photos from noise, keeping the paintings to be generated afterwards, starting from the photos' distribution which is ensured to be obtained from the initial noise distribution.

As far as the reliability of the MiFID score is concerned, this metric is destined to extract features from an intermediate layer and exploits the FID's strengths, which refer to capturing both quality and diversity at the same time; the FID score in turn bases itself on modeling the target alike and the generated images as Gaussian distributions. In addition to the FID score, MiFID score takes into account the tresholded minimum cosine distance across all training samples in the

feature space, averaged across all user generated pieces. A competitive metric could have been the structural similarity index measure (SSIM).

The paper logic is structured as follows. Chapter I presents the theoretical background which supports the implementation of generative adversarial networks within both solutions and the integration of Stable Diffusion within the second proposed solution. Chapter II justifies the synthesis of techniques employed by both solutions, showing the fundamentals which determined the achievement of a high score on Kaggle. Chapter III describes the solutions in detail, providing their architecture and proving their efficiency. Chapter IV gives the interpretation of the results obtained within the previous chapter, by making use of experimental comparisons and tabels. The last chapter matches the results with a meaningful conclusion.

Chapter 1

Deep generative modeling in theoretical terms

Both generative advesarial networks and Stable Diffusion fall into the category of latent variable models [3]. To summarize this type of models in mathematical and statistical terms, they firstly sample latent variables, $z \sim p(z)$, and then generate data objects, under the name of "observables", with the following distribution: $x \sim p_\theta(x|z)$. As it can already be inferred, z is a crucial factor in x's generation, a highly influential element or summary of elements describing the final image (e.g. the horse color in a picture of a horse or the silhouette of that horse). Naturally, through the use of the multiplication theorem, $p(x, z) = p(x|z) \cdot p(z)$. Since x is the only variable that can be accessed and sampled via training data, z has to be marginalized out:

$$p(x) = \int p(x|z) \cdot p(z)\, dz \tag{1.1}$$

However, decoding the above equation issues computability problems, due to the fact that the integral is — most of the times — not tractable. The base solution to this issue is represented by variational auto-encoders, which rely on the so-called variational inference [3], which defines a family of variational distributions, carrying a parameters set ϕ, $q_\phi(z)_\phi$. For a Gaussian distribution, ϕ would be viewed as $\phi = \{\mu, \sigma^2\}$. The core assumption regarding these distributions is that they assign non-zero probability mass to all $z \in \mathcal{Z}^M$, where M is the dimensionality of the latent space. This allows the following inference to take place, corresponding to the logarithm of the previously

stated intractable marginal distribution:

$$\begin{aligned}
\ln p(x) &= \ln \int p(x|z) \cdot p(z),\ dz \\
&= \ln \int \frac{q_\phi(z)}{q_\phi(z)} \cdot p(x|z) \cdot p(z)\, dz \\
&= \ln \mathbf{E}_{z \sim q_\phi(z)} \left[\frac{p(x|z) \cdot p(z)}{q_\phi(z)} \right] \\
&\geq \mathbf{E}_{z \sim q_\phi(z)} \ln \left[\frac{p(x|z) \cdot p(z)}{q_\phi(z)} \right] \\
&= \mathbf{E}_{z \sim q_\phi(z)} \left[\ln p(x|z) + \ln p(z) - \ln q_\phi(z) \right] \\
&= \mathbf{E}_{z \sim q_\phi(z)} \left[\ln p(x|z) \right] - \mathbf{E}_{z \sim q_\phi(z)} \left[\ln q_\phi(z) - \ln p(z) \right],
\end{aligned} \tag{1.2}$$

where Jensen's inequality was applied for solving the fourth transition. This result is followed by applying amortization over $q_\phi(z)$ and therefore replacing it with $q_\phi(z|x)$:

$$\ln p(x) \geq \mathbf{E}_{z \sim q_\phi(z|x)} \left[\ln p(x|z) \right] - \mathbf{E}_{z \sim q_\phi(z|x)} \left[\ln q_\phi(z|x) - \ln p(z) \right]. \tag{1.3}$$

The probabilities $p(z|x)$ are known as amortized variational posteriors and their ensemble justifies the name of the entire method currently being discussed. Amortization is employed into the original result so that a mapping between the latent and the observable variables is integrated in the shape of a distribution, facilitating the calculus of the Evidence Lower BOund (**ELBO**), which is the right-hand term of the inequality and, as will further be revealed, the objective function of the neural network to be built. The first part of this objective is often referred to as the (negative) reconstruction error, due to the process it describes: after x is encoded as z, it has to be decoded back to x. While measuring the difference between the learned latent variable and the prior distribution, the second member is more of a regularizer, which can be expressed with the help of the Kullback-Leibler divergence. To sum up, while $q_\phi(z|x)$ approximates the encoding of x into z in a stochastic manner, the conditional $p(x|z)$ identifies as the stochastic decoding of z into x. Both the encoder and the decoder are portrayed as neural networks and their parameterization happens as a consequence. In simple terms, as far as the working flow of variational auto-encoders is placed under discussion, the encoder and the decoder actually output the mean and variance of the distributions they represent; the parameters outputted by the encoder are then used to sample z and the samples are fed into the decoder.

The short description of the VAEs' functioning serves as a landmark for the way GANs came into being and the operating principles that kept them alive until now, even though there is still

a tiny dose of mistery surrounding them. The first difference they yielded was the capability to perform implicit modeling, instead of explicit modeling — which is specific to VAEs — meaning to get rid of the Kullback–Leibler and the prescribed distributions, which are both direct means for computing the difference between the target and the generated data. Instead of using the Kullback-Leibler divergence, the loss function is depicted as a neural network as well and its parameterization is learned during training; instead of outputting entire distributions, a single point is returned, as in the Dirac's delta function [3, 4]. In order to understand this behaviour, let us start with a perspective over the marginal distribution: the marginal distribution may be viewed as an infinite mixture of delta peaks, which leads us to the definition of implicit modeling, stating that once we generate an infinity of peaks from sampled z latents, we practically cover more and more regions of the observable space. In the following, we make use of an art related story to explain the principles guiding the way GANs can (re)create art pieces with the help of adversarial loss.

Imagine a plot having an art expert and a con artist (a fraud) as its main characters. As a side character, a worshiped real artist (such as Claude Monet) has passed away a while ago. The fraud's "job" is to mightily try to imitate the style of the real artist, whereas the job of the expert is to distinguish his work from the authentic pieces of the original artist. Paradoxically, we can think of the two main characters as working together: as time goes by, the expert will determine the fraud to improve his style and get closer to the original art, while the fraud will influence the expert to look at the fake paintings with a more critical eye and sanction them accordingly. Eventually, if this scenario happens within *the right parameters*, the pieces of the fraud will indistinguishably resemble the artist's work and the expert will be confused about his real/fake separation criteria. If we are to formalize this game, the fraud turns itself into a neural network playing the role of a *generator* and the critic transforms into a neural network which appears as a *discriminator* comparing details originating from real and fake art. As in the real world, the generator creates art from "nothing", which in this case translates as noise. The discriminator outputs the probability of an art object to be authentic. The two data sources, the real — the artist — and the fake one — the fraud — generate different data distributions. Considering

$$x \sim p_{data}(x) \tag{1.4}$$

as the real data distribution and

$$x \sim p_\theta(x) = \int G_\beta(z) \cdot p(z)\, dz \tag{1.5}$$

as the fake data distribution, both wrapped under the discriminator's classifier, the following objective function can be deduced:

$$\mathcal{L}(\alpha, \beta) = \mathbf{E}_{x \sim p\_real} \left[\log D_\alpha(x)\right] + \mathbf{E}_{z \sim p(z)} \left[\log \left(1 - D_\alpha(G_\beta(z))\right)\right]. \tag{1.6}$$

Consequently, the generator and the discriminator have two opposite objectives — which is the reason for calling it adversarial loss — leading the optimization to solving a min-max problem: the discriminator will try to maximize the objective function with respect to α, which is the discriminator's parameter, whereas the generator will tend to minimize the objective function with respect to β, which is the generator's parameter [Fig. 1.1].

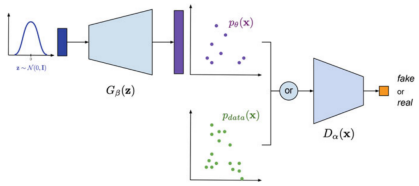


Figure 1.1: The general workflow of GANs. Please notice the similarity between the generator and the formal encoder presented in the short description of VAEs.

Lastly, Stable Diffusion models — which belong to the diffusion-based deep generative models category — take inspiration from hierarchical VAEs and aim at synthesizing images from noise, while walking through a prescribed number of timesteps [15, 16]. The time is looked at as a linear space in the vanilla version, serving as support for gradually turning the original images into noise and train a model (e.g. U-Net) to predict each noise increase. Therefore, the backward process within the training procedure is assumed to help the model's network learn the noise quantity added at some specific timestep — the timesteps being randomly sampled from a uniform distribution during training — while the forward procedure intervenes only for sampling in the reverse direction — imaging from random noise — after the training is finished. As the mathematical background is quite heavy, we won't dive into calculus details, as it was decided for this paper to keep the general overview of the progress from VAEs, to GANs and diffusion models, in the spotlight for this chapter. As a consequence, the theoretical improvement Stable Diffusion models are assumed to contribute consists of looking at variational posteriors as noise quantities added for each timestep

and use them to reverse the "noising" process. This models chain modifications applied to a Gaussian distribution, keeping conclusiveness by producing a Gaussian distribution in each time layer, including the last one, which prevents the posterior collapse problem [3].

Chapter 2

Literature review for experimental landmarks

All the solutions proposed within this paper rely on the experimental background of Cycle-Consistent Adversarial Networks (CycleGANs), proposed by J. Zhu, T. Park, P. Isola and A. Efros in [5]. They built the architecture for the purpose of unpaired image-to-image translation and in turn relied on the *pix2pix* framework belonging to Isola et. all and presented in [6], which attempted to perform image-to-image translation through the use of conditional generative adversarial networks. The idea of cycle consistency loss, computed in parallel by two different cross-domain generators, made CycleGANs, together with their various side improvements (such as TraVeLGAN, which adds a siamese network on top of the cyclic gnerator-discriminator architecture, with potential to produce an equivalent effect to that generated by the cycle consistency loss), still represent the current standard and one state-of-the-art involving GANs (and no pretraining) in the field of unpaired image-to-image translation [7].

If we are to decouple the Kaggle competition's statement from the generation problem itself, the new ITTR (unpaired Image-to-Image Translation with Transformers) framework, published in 2023, performs one-sided translation with a hybrid perception block and dual pruned self-attention [8]. Moreover, the CLIP TraVeLGAN architecture replaces the TraVeLGAN's siamese network with a contrastively pretrained language-image model; without thorough pretraining at the CLIP level, the corresponding research study reveals that the model is not able to perform the image-to-image semantic mapping [9]. Regarding the problem of image generation resembling the particular style of an artist, without any other constraints related to switching between two

domains or employing the GAN architecture, other state-of-the-art approaches also rely on text encodings and transformers. Even the Cycle Diffusion framework, which rests closer to the traditional CycleGANs, requires pretraining on text-to-image mappings in order to be able to perform image-to-image translation afterwards [10]. However, although these methods demonstrate higher potential than the previous GAN-based ones and require less parameters to be in place, all of them are supposed to take a lot of time to be trained from scratch and, therefore, to become more efficient than CycleGANs in terms of imaging quality, unless using some pretrained checkpoints. Nevertheless, a bigger dataset than the one containing only 300 Monet paintings would have definitely represented a critical requirement. The major improvement transformer-based layouts could have added to the Monet-style paintings generation task consists of being able to perform minimal changes where necessary — especially where the photo itself is quite close to a Monet-style piece of art — instead of deciding for none, as shown for some photographs in Fig. 3.12.

As far as the Kaggle competition's public scores and solutions are concerned, the first place is ahieved by employing the CLIP TraVeLGAN architecture, as it cand be inferred from reading the team's name and its members, who contributed to the paper proposing the CLIP TraVeLGAN model and described their Kaggle achievement within their research [9]. Regarding the code solutions investigated, the most notable results — if we are to go beyond the last three months which are summarized within the leaderboard — came from implementing the idea of the "CycleGAN with Better Cycles" paper, which proposes slight modifications within the calculus of the discriminator's loss, by weighting both the pixelwise and the feature level contributions within the cycle consistency loss [11]. This solution achieves a MiFID performance of 36.69193. Another solution worth being taken as baseline uses a base Stable Diffusion pipeline, with the additional mention that the images also needed to be upscaled within this solution, taking into account that the original samples are generated to be half the requested dimension, due to time constraints (as the Stable Diffusion forward process is prohibitively time consuming). This solution exceeds the competitions's alloted GPU time, since good results start showing up after the 100th epoch. Although the notebook does not provide an associated score, a personal attempt to replicate it scores around 109. During the next chapter, it will be shown that the second proposed solution aims exactly at improving this original baseline. Unlike the original baseline, the proposed solution implements Stable Diffusion from scratch for knowledge acquiring purposes [12].

Focusing on the experiments implying both solutions proposed within this paper, there are

three prevalent research landmarks which visibly influenced the results: *A U-Net Based Discriminator for Generative Adversarial Networks*, by E. Schonfeld, B. Schiele and A. Khoreva, and *Improving the quality of image generation in art with top-k training and cyclic generative methods*, by L. Vela, F. Fuentes-Hurtado and A. Colomer [13, 14]. As the title might suggest, the first article implements the U-Net architecture at the discriminators' level, with a twist, as CutMix strategy is also included and two losses are extracted from the network. The second article alters the generator's loss, by selecting only competitive individual per batch losses to take part in the backward process. Taking into account that the progress in science and, therefore, in the target field of image-to-image translation, implies finding new solutions and not only blindly repeating what others have already created, the intention of the experiments described within this paper was to not reproduce any of the already submitted notebooks, unless it significantly improves them.

According to the results of a previous prizes round of the competition, the first proposed solution within this paper would have gained a silver medal under the same conditions.

Chapter 3

Proposed solutions

The two solutions proposed within this paper have distinct purposes and try different incremental improvements for the current well performing models in the target area. The first solution is the one which was destined to be evaluated within the Kaggle competition announced in the previous chapters, while the second one tries to outperform previous Stable Diffusion attempts to generate Monet-style paintings, in terms of the MiFID score and the intelligibility of the depicted objects, this time while involving the provided photographs as well. This solution implements a different understanding of the GAN framework. Both main solution ideas are then subject to different variations within the ablation study sections, in order to measure their robustness and report the success or failure experimented with further attempts of potential improvement.

3.1 First solution: Improved CycleGAN

3.1.1 Main architecture

As it can be observed in Fig. 3.1 and Fig. 3.2 – picture (I), the generator consists of a U-Net holding upsambling and downsampling blocks, with normalization layers in-between and activation via the ReLU function, whereas the discriminator embodies a stack of downsampling layers, in order to perform feature extraction up to a 15×15 representation, as specific to the PatchGAN thesis [6]. To dive into more details regarding the generator, the output of the downsampling layers is concatenated to the output of the connected upsampling layers, but the last layer belonging to each of the two paths (the downsampling path and the upsampling path) makes exception from this rule. As a general prescription, the kernel size of the downsampling/upsampling convolutions

is set to 4, the stride is fixed at 2 and the padding is set to 1. Apart from particular batch normalization and dropout regularizations, the key difference between the two paths' components consists in performing a classical convolution for the downsampling components and a transposed convolution for the upsampling ones.

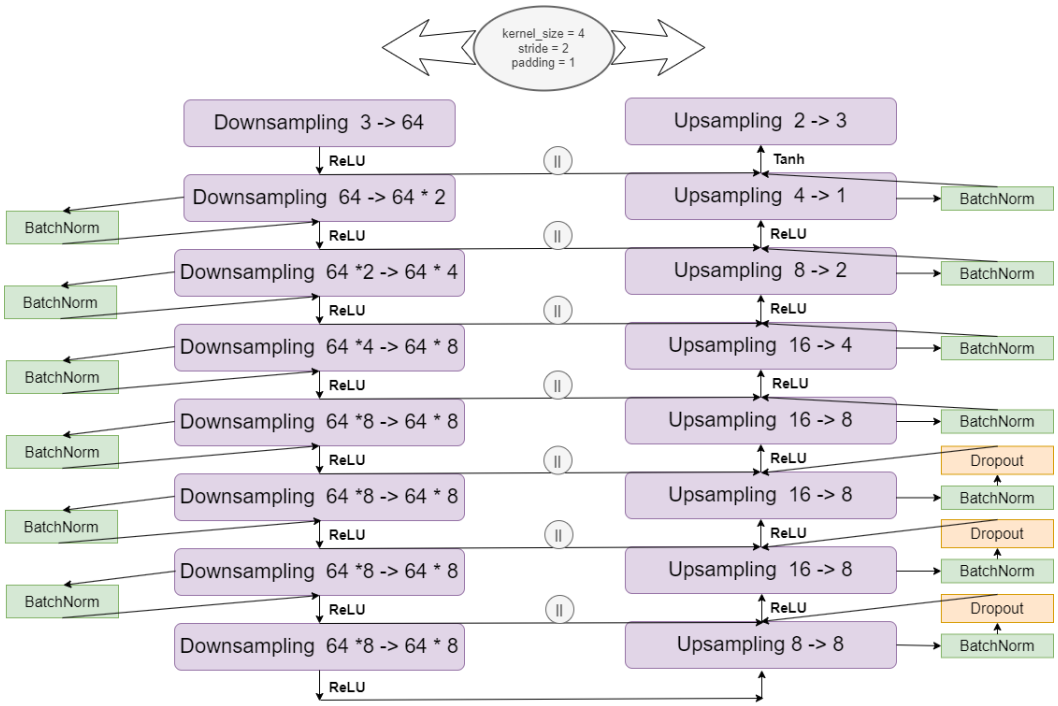


Figure 3.1: First solution – improving CycleGAN: generator's architecture.

The main idea of the CycleGAN implementation specific to this art recreation task is to train photo-to-painting and painting-to-photo generation simultaneously. Therefore, two identical generators following the architecture depicted in Fig. 3.1 are introduced within the CycleGAN model and two discriminators following the structure in Fig. 3.2 are needed for deciding over the paintings and the photos, respectively. At each training step, the forward process imposes a painting to be sampled from a photo and a photo to be sampled from a painting. Moreover, as the loss formulas will motivate, the paintings are also fed into the photo-to-painting generator and, simetrically, the photos are fed into the painting-to-photo generator as well. Lastly, paintings are reconstructed from the fake photos sampled in their turn by feeding a painting into the painting-to-photo generator; simetrically, photos are reconstructed from the previously sampled fake paintings. As far as the backward step, precisely the loss implementation, is concerned, there are three different losses implied by the generator: the well-known adversarial loss, the identity loss and the cycle loss. All three are implemented with the same parameters for the two generators, concerned

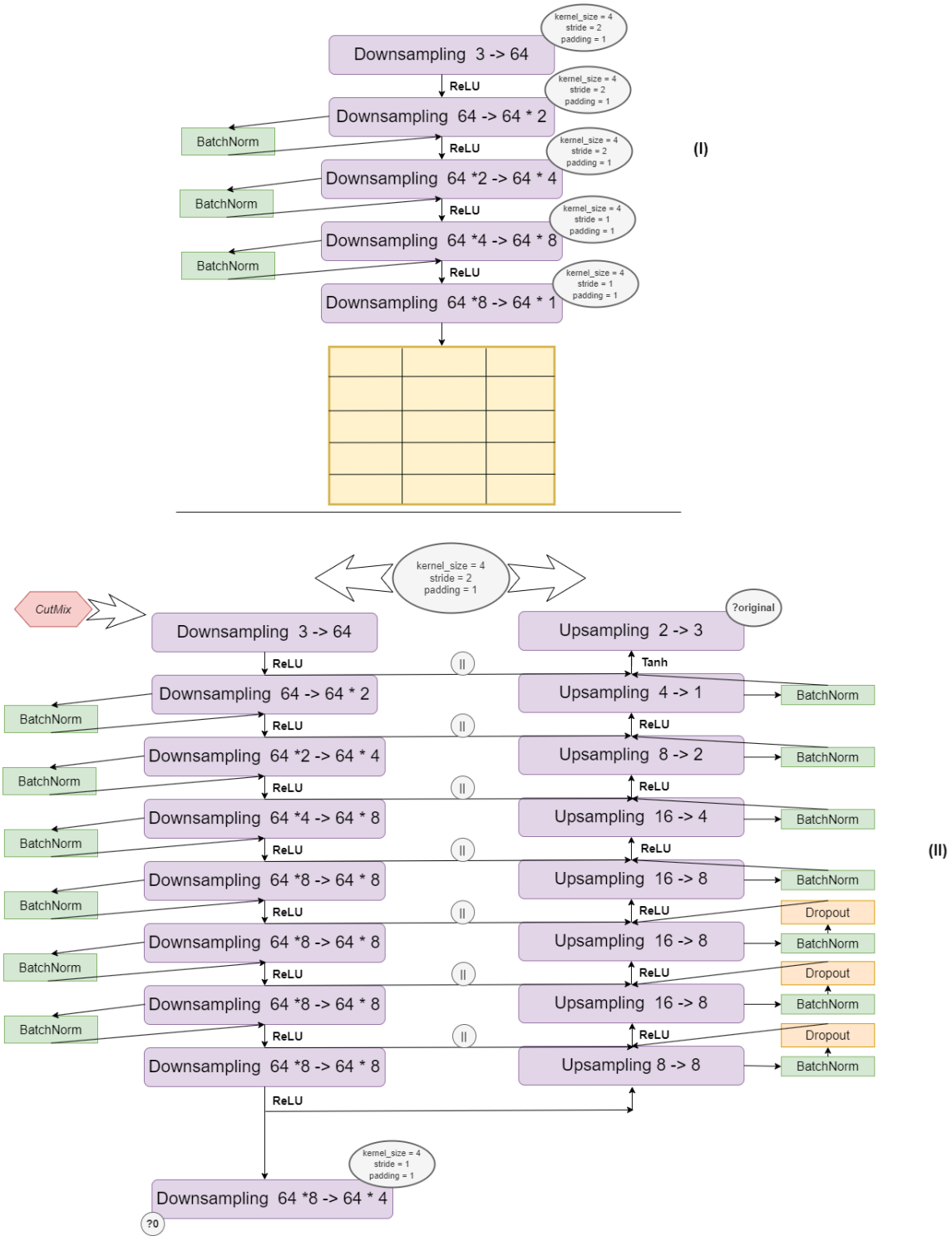


Figure 3.2: First solution – improving CycleGAN: discriminator's architecture.

with images and photos, respectively. The adversarial loss measures the difference between the generated image/photo and a tensor of the same shape, filled with ones; the significance of closer to 1 values within the discriminator's output corresponds, by convention, to a higher probability that the analyzed piece is authentic; on the contrary, values closer to 0 indicate a high probability of fraud. The identity loss checks the compatibility between the output of feeding the photo-to-painting generator with paintings and the original paintings given as input to the same mentioned generator. They should be as close as possible to each other, the mission of the identity mapping being to preserve the input's colors within the output, similar to the neural style transfer technique, which does not rely on learning nuances, but focuses exclusively on style transfer. Simetrically, the output of feeding the painting-to-photo generator with photos is compared to the original photos. The cycle consistency loss compares the real and the reconstructed pieces, evaluating full connected loops across the two generators. The L1 loss was used for both the identity and the cycle loss, whilst the mean squared error was employed for the adversarial losses. As it might have already been intuited, the adversarial loss at the discriminator level translates into learning the classifier to distinguish the real pieces as being authentic and the ones generated by the first two forward calls as fake. The discriminator outputs patch classification, which means that it sequentially relies on the features within subsections of the photo/painting and that gives it the power to neither be too generic and overfit too fast (if the comparison would have been performed on a single scalar output), nor become too specific (if the comparison would have been performed pixelwise). Obviously, the real pieces are compared with a ground truth consisting only of ones, while the fake pieces — as opposite to the generator's adversarial loss — are compared with a ground truth filled with zeros. Regarding the comparison process of the fake pieces, a randomized buffer consisting of past and current images is employed so as to cover longterm evolution and more generic and stable behaviour; with a probability of 0.5, the currently generated pieces will replace the previous ones already located in the buffer. The application of the image buffer is to motivate the classifier to keep learning from the past as well and to give a chance to the past generations to blend with the more actual ones.

Having the algorithmic skeleton in place, its efficiency — as specific to the generative adversarial framework —- ultimately depends on balancing the influences coming from all of the components. The model parameterization is rather unstable and demands for very high precision with respect to each ratio involved in the learning process. The specific parameters to be decided

at the finetuning stage are: the identity loss ratio in the generators' loss, the cycle consistency ratio in the generators' loss and the ratio for combining the two adversarial losses (with respect to the real and the fake piece) within the discriminators, the batch size and the image buffer size. Apart from these, the learning rate, the number of epochs after which it starts to decay, the optimizer itself and the optimizer's parameters, such as the "betas" for Adam, need to be set to a favorable value as well.

Building on the edge of the previously described base structure, this paper adds a suite of potential improvements to the solution and evaluates their influence on the results in the ablation study section. Therefore, taking inspiration from [13], Fig. 3.2 – picture (II) depicts a U-Net discriminator architecture relying on CutMix augmentation. The real and the fake piece are firstly combined using CutMix (or MixUp, but for this research CutMix was used). After traversing the downsampling path, the mixed image is evaluated with respect to its overall authenticity, which is considered to pass as fake due to the CutMix augmentation which will always imply an influence from the fake piece. Reducing the complexity of the U-Net might result in a patch classification at this level, but doing so did not result in any quality improvements on the particularly studied dataset. The final output of the U-Net network is pixelwise compared to the raw mixed piece, generating a very specific output, in contrast with the quite generic one extracted from the middle. The same article proposed the regularization consistency loss, which may be added to the generator's loss and essentially targets the difference between the real and the fake piece by making use of the CutMix augmentation: poviding that $mask$ is the CutMix binary mask and $fake$ and $real$ are the pieces entering the loss computation, the regularization consistency loss calculates the L_1 distance to evaluate the gap between the two:

$$L_1(mask \cdot real + (1 - mask) \cdot fake, (1 - mask) \cdot real + mask \cdot fake). \tag{3.1}$$

Needless to say, the introduction of this new loss adds a new parameter to the model, namely the regularization consistency loss ratio within the generator's loss.

Another potential improvement consists in top-k taining, which implies the backward process for the generator's loss to account only for the most influential images in the batch, in other words for the images which have the higher contribution to decreasing the loss. This is implemented through a sorting algorithm which selects only the first k images having the lowest individual losses. As the first epochs usually generate noisy pieces, there is no need to implement this technique during the first steps, as it would not make a real difference. However, it is natural for k

to have a higher value during the early epochs and decrease as the influences are better outlined, as long as an empirically established treshold for decreasing is not exceeded; a recommended value, the same as the value decided by this paper's implementation, is 0.75% of the batch size.

The last improvement idea came from the "CycleGAN with Better Cycles" paper and consisted in extracting the lower level features from an intermediate layer of the discriminator and balancing their contribution with that of the outputted higher level ones [11]:

$$L_{cycle} = cycle\_ratio \cdot (\gamma L_1(real, recon) + (1 - \gamma) L_1(F(real), F(recon))). \tag{3.2}$$

where *recon* represents the reconstructed image and F is the feature extraction network (a subordinate downsampling block clipped from the discriminator). Again, this introduces a new finetuning parameter, γ, responsible with weighting the two contributions.

3.1.2 Ablation study and benchmark performance

This section studies the influence of modifying different architectural details and parameter values over the MiFID score, time, stability and over the natural visual quality of the image, as perceived by the human eye. It can be viewed as an evidence of the progress made within the project's implementation. The measured scores also depend on the stage of optimization at the moment the evaluation of a particular aspect was performed, but additional evidence has been gathered in the subsequent runs they are still topical for the problem considered anyway. Limited computational resources need to be taken into account as well, since the training process had to fit in the time alloted for the notebook to run by the Kaggle competition's policy.

As Table 3.1 illustrates, the batch size has a crucial influence over the results. Given the fact that all the runs for the first solution were scheduled to fit in 4 hours and 55 minutes of GPU time, a larger batch size unfortunately implies less training steps, which means less weight updates and, consequently, worse results. Even though the top-k training improvement slightly increases the score, it is not enough compared to insufficient updates during training. The question if longer training time would determine the configuration to perform better on larger batches remains open, as a very large amount of GPU time would have been necessary to answer that. During the experiments performed, a batch size of 1 proved to be optimum for the target running time.

As Table 3.2 and Table 3.3 express, the U-Net discriminator does not perform as well as the classical feature extraction discriminator, the visual results appearing to be noisy and depicting

Table 3.1: First solution: MiFID score comparison based on the batch size.

Criterion	**Batch size of 1**	**Batch size of 32**	**Top-k training: batch size of 32**
Number of epochs	17	50	50
Number of steps	119646	43950	43950
MiFID score	39.87871	50.21598	49.25094

regular patterns, which is not the case if we take an overview of the Monet's paintings. Moreover, by analyzing Table 3.4, it can be observed that the adversarial losses proposed in the original discriminator have to be kept in the U-Net discriminator as well in order to achieve a higher score, although the research study which came up with the second version did not include the classical loss functions in its implementation.

Table 3.2: First solution: MiFID score comparison based on the U-Net discriminator architecture.

Feature extraction architecture	**U-Net architecture**
39.87871	59.4159

Table 3.5 and Table 3.6 show the importance the identity loss exerts over the generated results. A larger weight for this loss within the total generation loss results in keeping the general tendency of the original colors, whilst a smaller weight changes the color palette of the picture and causes penalties with respect to the MiFID score. The value of the identitiy loss ratio leading to the highest overall MiFID score was 10, the same as the value employed for the cycle loss.

As it can be observed in Table 3.7, post-processing data for noise reduction or contours sharpening pulls down the score instead of lifting it up as it might have been expected; this can be viewed as a sign that the model picks up exactly what it needs from Monet's style and does not alter the specifics during translation, therefore the images outputted by the generative adversarial network have to be kept intact.

Table 3.8 shows that the current model's architecture is not subject to gaining improvements if the equations from [11] are implemented within the cycle loss function.

As clearly illustrated by Table 3.9, larger buffer sizes result in higher MiFID scores, which proves the efficiency of keeping generated paintings for more than the end of the iteration they show up in. A suitable value for the buffer size proved to be 100. The first experiment in the table took 10 as the buffer size value.

Table 3.3: First solution: Visual results comparison based on the U-Net discriminator architecture.

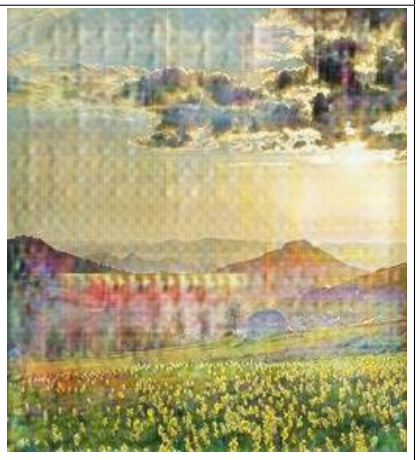
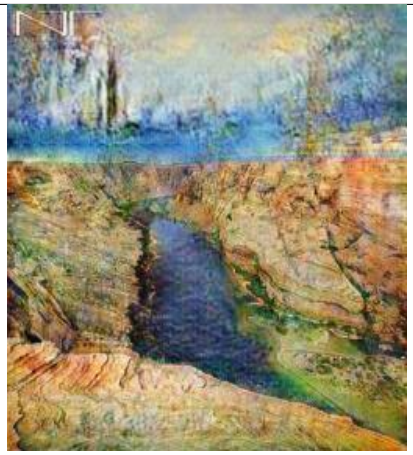
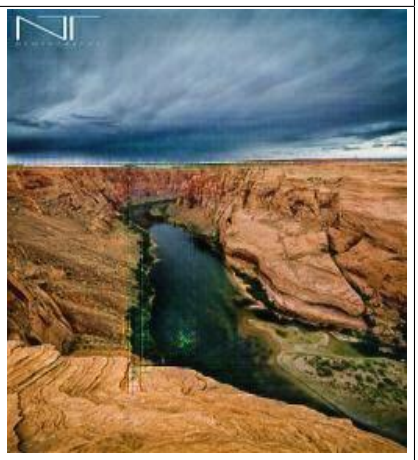
Feature extraction architecture	**U-Net architecture**
	
	
	

Table 3.4: First solution: MiFID score comparison based on the U-Net discriminator losses.

U-Net plus classical discriminator losses	**U-Net as in source [13]**
59.4159	61.73151

Table 3.5: First solution: MiFID score comparison based on the identity loss.

With color preservation (higher ratio)	**Without color preservation (lower ratio)**
42.49448	43.02967

Table 3.6: First solution: Visual results comparison based on the identity loss.

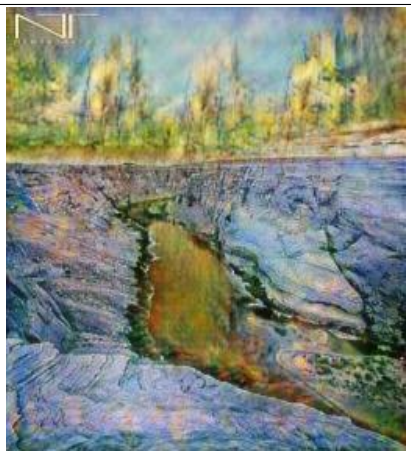
With color preservation (higher ratio)	**Without color preservation (lower ratio)**
	
	
	

Table 3.7: First solution: MiFID score comparison based on data post-processing.

With data post-processing	**Without data post-processing**
45.05926	42.49448

Table 3.8: First solution: MiFID score comparison based on the cycle loss.

With better cycles	**Without better cycles**
44.99861	42.49448

A new improvement proposed for the algorithm was the *output-as-input* technique, the personal contribution which atually determined the last edge increase in the highest score within this research to be achieved. This contribution consists of splitting the training process in two parts: the first one stays the same, but integrates a smaller number of epochs, whereas the second takes the output from the previous as input for the photos. Assuming the previously obtained photos denote pretty high quality, as much as possible given a lower number of epochs, the second stage makes use of the closer relationship between them and the original paintings.

As depicted by Table 3.12, there are several photos (almost) left untouched by the inference. As a general observation, it seems that these photos correspond to the cases when unknown elements dominate the picture; for instance, bricks do not appear in Monet's paintings at all.

3.2 Second solution: Stable Diffusion embedded in GAN

3.2.1 Main architecture

As it can be observed in Fig. 3.5 and Fig. 3.4, with their more detailed cross section analysis in Fig. 3.6, this solution may keep one of the discriminators employed by the first solution and focus on the generators' architecture. The CycleGAN proposition of having two generators is also continued, except that this time the generators are represented by Stable Diffusion models

Table 3.9: First solution: MiFID score comparison based on the buffer size.

Smaller buffer size	**Larger buffer size**
59.93715	49.25094

Table 3.10: First solution: MiFID score comparison based on the output-as-input technique.

With output-as-input	Without output-as-input
39.87871	42.49448

Table 3.11: First solution: Visual results comparison based on the output-as-input technique.

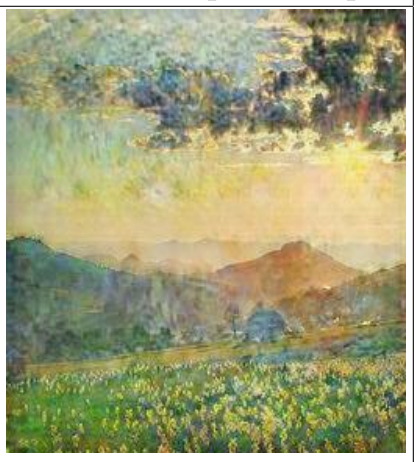
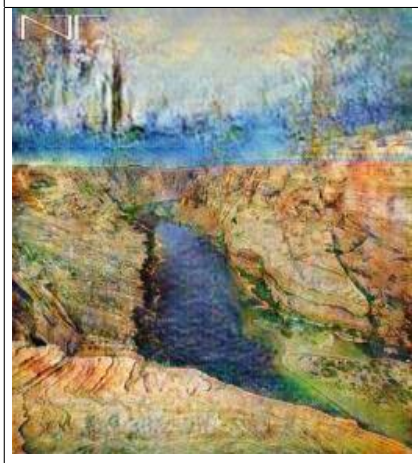
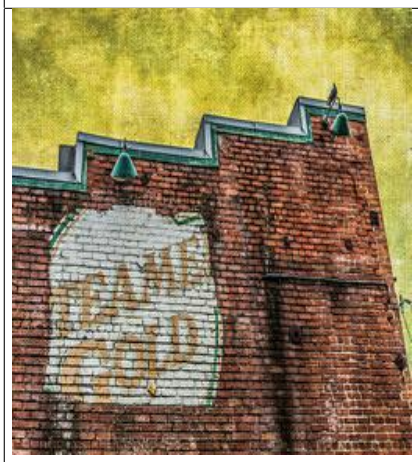
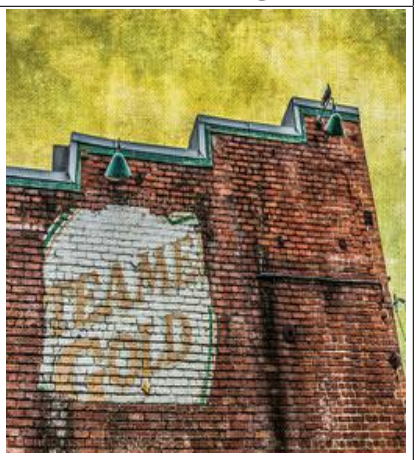
With output-as-input	Without output-as-input
	
	
	

Table 3.12: First solution: Apparently untouched photos.

Photo	Painting
	

built upon double U-Net neural networks. The main idea of this new architecture is to create a collaborative and adversarial — both at the same time — environment mostly through the generators' architecture itself. Therefore, the purpose of the double U-Net model is to create a chain such as the one depicted in Fig. 3.3, so that, through the power of following the evolution of the same distribution, the paintings and the images put together their style and structure. This results in quicker contours drawing and artificially enlarges a small dataset, such as the one consisting in Monet's paintings exclusively and employed as it is by the vanilla Stable Diffusion model whose improvement is under discussion. Starting from noise, both generators approach their goal (photos or paintings) through the intermediate generation of the opposite goal (paintings or photos), which is ensured to be made part of the same distribution. As the training advances, photo drawing as a target implicitly encourages paintings drawing until the middle of the chain, when the roles are switched and the paintings serve as a base for generating photos (Fig. 3.4); the same happens within the second generator, where drawing paintings implicitly encourages drawing the randomly pairing photos until the middle of the chain, when the created photos themselves start to encourage drawing paintings (Fig. 3.4).

As far as the backward process is concerned, since the generators create the adversarial environment themselves, the discriminator was discarded in the metrics evaluation due to limited resources. However, if we were still to employ it, it would have performed the same calculus as described in the case of the first solution, but having the fake photo generated there replaced by the middle piece extracted from the generator aiming at the opposite goal. The key pinciple of

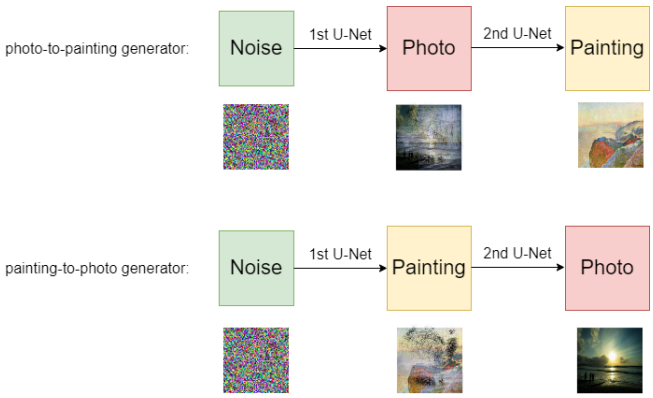


Figure 3.3: The general architecture and purpose of the two double U-Nets (generators) within the second solution.

the backward propagation is to avoid forward passes through the Stable Diffusion paradigm, since there were no resources available to achieve that. Even better, this can translate into a potential solution for avoiding such forward passes if they are found to be needed in future approaches. The fake pieces are represented by the middle result of the generator having the opposite goal. In the case of the paintings, the fake pieces will be the middle output — that is, the output of the first U-Net — of the photos generator, whereas the fake photos will be the middle output of the paintings generator. The loss specific to generators might keep the indentity and cycle functions having this new acception of fake pieces, but the choice of the evaluated implementation was to discard them as proceeded in the discriminator's case as well. Eventually, the essential components of the generator's loss is the difference between the induced noise and the noise predicted by the network for the sampled timestep — the classical noise of a Stable Diffusion model — calculated as mean squared error, and the difference between the noisy images sampled (the ones fed into the model) and the fake pieces extracted from the middle of the opposite network, calculated as $L1$ distance. The reason this might work is because the two networks are trained with the same sampled noise over the photos/paintings and, at a specific timestep, the middle piece should resemble the piece fed into the opposite network as much as possible. It is important to note the symmetry of the double U-Net model, which facilitate the logic of the forward step and the backward process to be consistent with each other. Obviously, for the prediction step the noise-photo-paining network will be used.

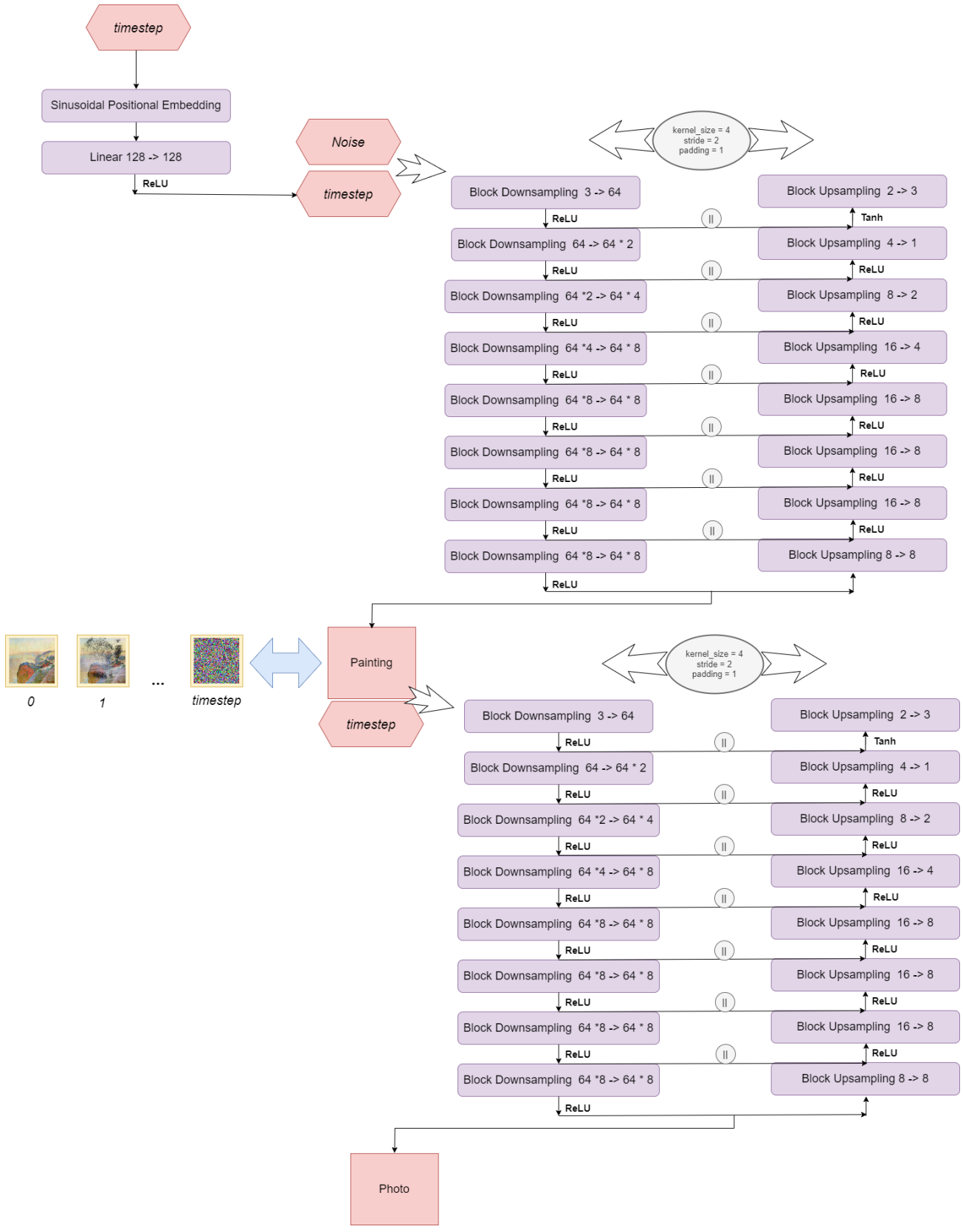


Figure 3.4: Second solution – GAN embedding Stable Diffusion: paintings generator's architecture.

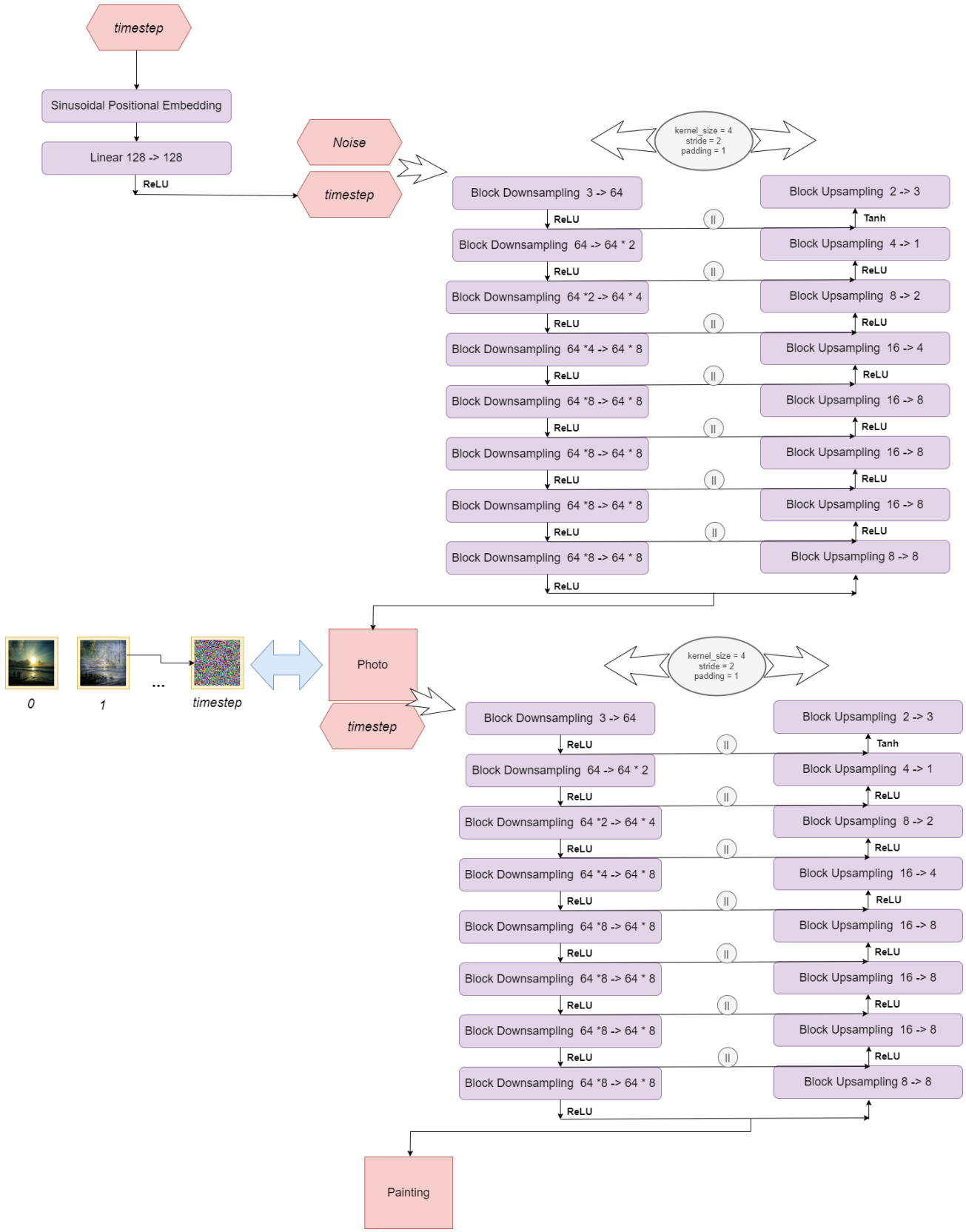


Figure 3.5: Second solution – GAN embedding Stable Diffusion: photos generator's architecture.

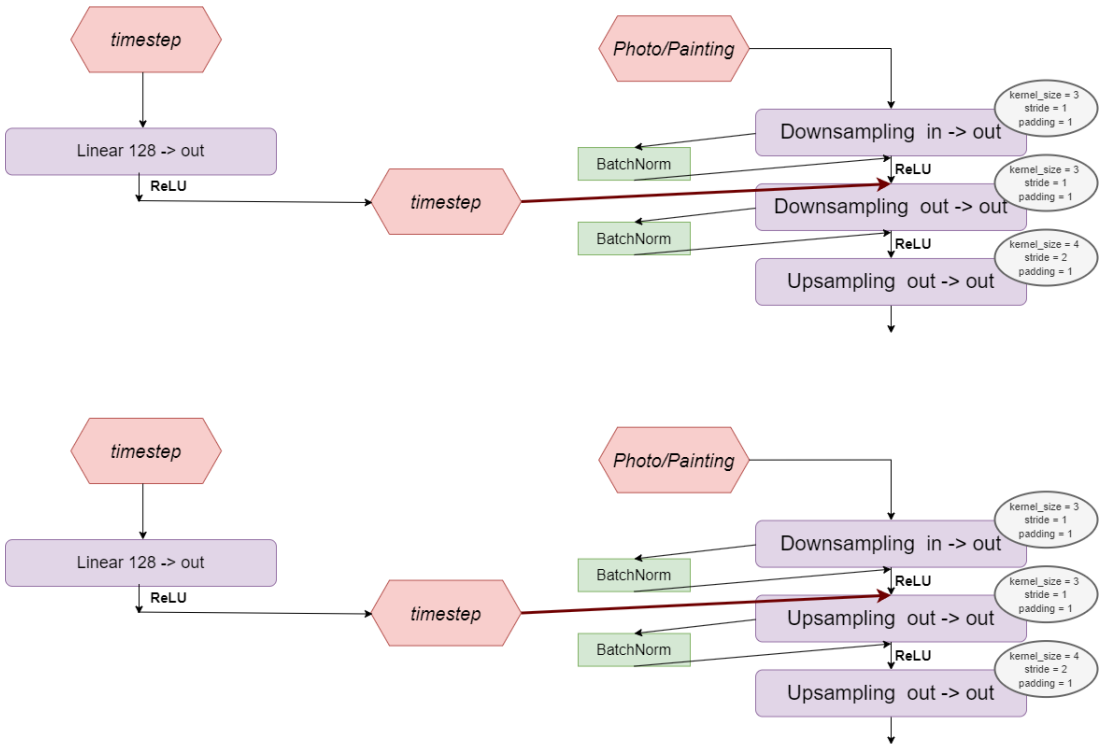


Figure 3.6: Second solution – GAN embedding Stable Diffusion: upsambling and downsampling blocks' architecture.

3.2.2 Ablation study and benchmark performance

The solution proposed within this part represents an ablation study itself, since it aims at proving that Stable Diffusion might perform better if integrated into a different kind of generative adversarial environment and does not propose to improve a close to state-of-the-art score, but rather to induce progress at the level of a currently weak result. Therefore, we refer to the analyzed metrics accordingly.

Since training only on the paintings collection induces only vague contours — which appear very slowly during the training process — on the created piece of art, the analysis below shows a visual quality improvement if details from the photos are integrated into te distribution leading to the creation of paintings [Table 3.14].

Moreover, the MiFID score increases with a quantum of 40 in less epochs of training (even though, we should take into account the dataset is also bigger now, so we actually measured the increase in score by allowing 5 hours of training for both experiments and as much as needed for inference) [Table 3.13].

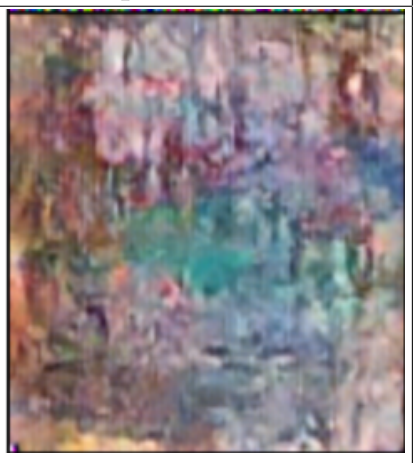
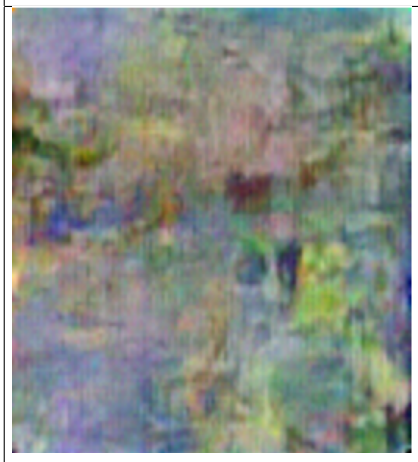
Further work might involve experimenting more with attention blocks at the seventh layer in

the downsampling path and the second layer in the upsampling path, as already started, but due to limited training time resources, only intuitive visual improvements can be recorded for now.

Table 3.13: Second solution: MiFID score comparison

Vanilla Stable Diffusion	**Proposed solution**
109.47919	69.90175

Table 3.14: Second solution: Visual results comparison

Vanilla Stable Diffusion	**Proposed solution**
	
	
	

Chapter 4

Final discussion on results

Apart from aiming at high scores with just right parameter values and granular optimization, it is also rewarding to come up with a significant upgrade to a weaker, vanilla solution, as it will count as a longterm, applicative discovery. While there is certainly room for experimenting with other ingenious pipelines and methodical parameter variations, the current solutions comprise a strong baseline for further developments. Noticing the general unstability of the GAN pipeline, the second solution has already eliminated some effort regarding the parameters setup. As long as the first solution is very sensitive with respect to different parameterization variants (which are a lot), progress on the topic will take small steps, which is why a new progress track might favorably begin with the second proposed solution, since it demands for less parameters to be decided upon.

The proposed dataset has also represented a challenge, due to a reduced amount of paintings to be paired with a much larger number of photos, thus affecting the diversity index within the MiFID score. Pairing the photo files and the paintings collection and developing two generators represent the CycleGAN framework's specific way of tackling this issue, whereas the second solution takes advantage of wrapping the pairs under the same distribution being evolved.

Walking through the training process on both solutions and out of 20 competition submissions (and another couple of runs on the personal machine), there does not appear to be a recipe for determining the checkpoint yielding the best score from a target interval of training steps. The generator's and the discriminator's losses sharply decrease during the first epochs, but then oscillate within quite small intervals of values, which makes it even harder to anticipate the potential of the solution before seeing the actual generated pictures. Nevertheless, the multitude of experiments performed demonstrate the need for longer training time in order to draw other further conclusions.

Chapter 5

Conclusion

Taking into account the benefits and limitations of each variant presented, we are able to understand why there are multiple approaches out there for a unique proposed problem to be solved. While generative artifficial intelligence has definitely taken the lead as one of the most spectacular advancements in the neural networks research field, there is a lot of effort to put in the configuration details of the generative framework the creator decides upon. The process of recreating the sensitivity that artists engrain into their work has its own high sensitivity in terms of parameters modification and training duration. Keeping the spirit of time efficiency where possible, simpler base architectures (i.e. U-Nets with base upsampling and downsampling blocks) have been integrated in conceptually loaded models (i.e. GANs) in order to be able to experiment a large number of setup variations. All in all, mimicking the colors and brush Claude Monet would have picked is a suitable topic for investigating additional metrics such as graphical connectivity and scalability, proving that emulating creativity lives somewhere beyond real-life examples comprehension, performed prior to exerting subjectivity, in the middle of the road leading from imitation to innovation.

Bibliography

[1] Kaggle, "I'm Something of a Painter Myself," Available: `https://www.kaggle.com/competitions/gan-getting-started` [Accessed: Jan. 16, 2024].

[2] Kaggle, "What is MiFID (Memorization-informed FID)?," Available: `https://www.kaggle.com/competitions/gan-getting-started`[Accessed: Jan. 16, 2024].

[3] J. M. Tomczak, "Deep Generative Modeling," Springer, February 2022.

[4] Arsen M. Shutovskyi, "Some applied aspects of the Dirac delta function," Journal of Mathematical Sciences, vol. 276, November 2023.

[5] J. Zhu, T. Park, P. Isola and A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), October 2017.

[6] P. Isola, J. Zhu, T. Zhou and A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[7] Y. Pang, J. Lin, T. Qin and Z. Chen,"Image-to-Image Translation: Methods and Applications," IEEE Transactions on Multimedia, vol. 24, January 2021.

[8] W. Zheng, Q. Li, G. Zhang, P- Wan and Y. Wang, "ITTR: Unpaired Image-to-Image Translation with Transformers," arXiv, March 2022.

[9] Y. Bodyanskiy, N. Ryabova and R. Lavrynenko, "CLIPTraVeLGAN for Semantically Robust Unpaired Image Translation," EasyChair Preprint no. 9933, April 2023.

[10] C. H. Wu and F. De la Torre, "Unifying Diffusion Models' Latent Space, with Applications to CycleDiffusion and Guidance," 2023 International Conference on Computer Vision (ICCV), October 2023.

[11] T. Wang and Y. Lin, "CycleGAN with Better Cycles," 2023 International Conference on Computer Vision (ICCV), University of California, Berkeley, Available: https://www.tongzhouwang.info/better_cycles/report.pdf [Accessed: Jan. 15, 2024].

[12] DeepFindr, "A Diffusion Model from Scratch in Pytorch," Google Colab, Available: https://colab.research.google.com/drive/1sjy9odlSSy0RBVgMTgP7s99NXsqglsUL?usp=sharing&fbclid=IwAR0XRrSE8T1pYZzPHnHPJUMMM3dlNUZoM_mJ89-mtMsNmEyWgX6zVaJiQQY [Accessed: Jan. 7, 2024].

[13] E. Schonfeld, B. Schiele and A. Khoreva, "A U-Net Based Discriminator for Generative Adversarial Networks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[14] L. Vela, F. Fuentes-Hurtado and A. Colomer, "Improving the quality of image generation in art with top-k training and cyclic generative methods," Nature Journal, no. 13, October 2023.

[15] DeepFindr, "A Diffusion Model from Scratch in Pytorch," YouTube, Available: https://www.youtube.com/watch?v=a4Yfz2FxXiY [Accessed: Jan. 7, 2024].

[16] Outlier, "Diffusion Models — Paper Explanation — Math Explained," YouTube, Available: https://www.youtube.com/watch?v=HoKDTa5jHvg [Accessed: Jan. 9, 2024].