UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA (UNED)
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

DETECTING OVERFITTING IN GANS WITH A METRIC BASED ON THE FOURIER SPECTRUM

TRABAJO DE FIN DE MÁSTER PRESENTADO POR
ÁNGEL JAVIER GAMAZO TEJERO

DIRIGIDO POR
MARIANO RINCÓN ZAMORANO
JOSÉ MANUEL CUADRA TRONCOSO

MÁSTER UNIVERSITARIO EN I.A. AVANZADA: FUNDAMENTOS, MÉTODOS Y APLICACIONES
CURSO 2019-2020
CONVOCATORIA DE SEPTIEMBRE

CONTENTS

DETECTING OVERFITTING IN GANS WITH A METRIC BASED ON THE FOURIER SPECTRUM

Javier Gamazo Tejero
Departamento de Inteligencia Artificial
UNED
agamazo6@alumno.uned.es

Mariano Rincón Zamorano
Departamento de Inteligencia Artificial
UNED
mrincon@dia.uned.es

José Manuel Cuadra Troncoso
Departamento de Inteligencia Artificial
UNED
jmcuadra@dia.uned.es

ABSTRACT

Recent progress in generative image modeling is leading to a new era of high-resolution fakes visually indistinguishable from real life images. However, the development of metrics capable of discerning whether images are synthetic or not runs behind the race of achieving the best generator, thus bringing potential threats. We propose a rotation invariant metric capable of distinguishing real and generated images and prove its performance and correlation with subjective evaluation on a brain MRI dataset to generate synthetic white matter lesion images. We name this metric CSD (Circular Spectrum Distance) due to its circular nature and its inherent relation to the Fourier Spectrum. We find that this metric, as opposed to Frechet Inception Distance or Inception Score, detects overfitting during training in terms of generator memorisation without making use of any pretrained network. The conclusions are generalized to CelebA-HQ as a benchmark dataset.

***Keywords* —** GAN metric, Fourier Spectrum, overfitting, memorisation

1 INTRODUCTION

Since their introduction, generative adversarial networks (GANs) (Goodfellow et al. 2014) have dominated the state of the art in image generation tasks. It has been demonstrated that GANs are capable of learning very different distributions, producing images that are indistinguishable from the real ones (Karras et al. 2019a), (Karras et al. 2019b). However, the question of how to assess their performance remains open. The subjectivity of this assessment, bound to the high variety of domains in which an image generator could be trained, makes the effort of developing a universal metric, valid for all image generation models, a tremendously hard task. Therefore, up to this day, and even though several quantification metrics have been proposed (Salimans et al. 2016), (Heusel et al. 2017), (Im et al. 2016), (Olsson et al. 2018), the most correct method of assessing image quality is through subjective evaluation, relying on humans to give their opinion on the level of reality of generated images.

The goal of developing an evaluation metric is to carry out this process in an objective, automatic way. The applications of such a metric are multiple. First, it can be used to compare generative models, providing a benchmark of image and model quality, and also helping in the assessment of new architectures, therefore facilitating the search for an optimal architecture. Second, so far there is no consensus on the conditions for early stopping in generative adversarial networks. A reliable metric would help in finding the "sweet spot" where the architecture has reached its optimal point. It requires to detect overfitting, either coming from mode collapse, which arises when the generator produces a given set of defined outputs that fool the discriminator, or when it produces data already in the dataset (memorisation). Third, the emergence of "deepfakes" is a growing problem, as has

been pointed out in several non AI specialized journal articles (Toews 2020), (Shao 2019). A metric could help to detect fake images or videos generated with Deep Learning methods.

Following (Borji 2019), an efficient evaluation metric should not only favor models that achieve high fidelity samples, agreeing with humans' subjective perception, but also be able to detect diversity in the generated samples, understanding overfitting, mode collapse and mode drop[1]. Also, efficient metrics should favor models with disentangled spaces or be sensitive to image distortions, such as rotation or pixel translation. Nowadays, the most used metrics can detect whether synthetic samples are close to reality, like Fréchet Inception Distance (FID) (Heusel et al. 2017) or Inception Score (IS) (Salimans et al. 2016), however they are coupled with big disadvantages, such as high computational cost or dependency on a pretrained model, that keeps them far from being universal metrics.

To the authors' knowledge, all the proposed metrics use either pixel space or latent space computed with a neural network. However, none of them has dug deep in the goodness of spectrum representations and the inherent difficulty for GANs to reproduce the high frequencies of real samples. In this work we propose a new metric for GAN evaluation based on the discrepancies found in previous studies in high-frequency characteristics for deep neural networks (S.-Y. Wang et al. 2019), (X. Zhang et al. 2019) and apply it to brain magnetic resonance images (MRI) generated with StyleGAN (Karras et al. 2019a). We propose a method that makes use of Fourier Spectrum images for both real and generated datasets to quantify their differences. This method is motivated by the metric features described in the previous paragraph and by the inherent rotation invariance present in MRI. This work contributes to the state-of-the-art with:

- An usage of Fourier transform to evaluate the performance of the generator in a GAN
- The proposal of a metric that can detect overfitting in terms of memorisation in GANs and correlates positively with subjective evaluation
- A metric that follows FID behaviour during training but needs smaller batch sizes and thus is faster to compute

The following sections of the paper are organized as follows. In Section 2 we give a brief overview on the state of the art of GANs, their evaluation metrics and an introduction to the use of complex numbers in artificial vision, in Section 3 we detail the foundations of the new metric based on Fourier transform. Section 4 presents the GAN architecture we have chosen to prove the metric. Finally, Sections 5 and 6 will show the results and conclusions respectively.

2 Related work

2.1 Generative Adversarial Networks

2014 marked a major breakthrough in the field of Computer Vision and image generation with the first proposal of Generative adversarial networks (GAN) (Goodfellow et al. 2014). This type of architecture consisted of two models (mainly multilayer perceptrons) who worked together in a cooperative way trying to find the solution of a minimax game. The generative network was confronted with an adversary whose objective was to discriminate between samples of real and generated distributions.

The training of a basic GAN as described in (Goodfellow et al. 2014) to generate new samples from a given distribution of data $p_{\text{data}}(x)$ starts with a random distribution $p_z(z)$ that is then mapped to data space (p_g) using a multilayer perceptron (generator) denoted by $G(z;\theta_g)$, where θ_g refers to its parameters. A second perceptron, $D(:;\theta_d)$, is added with the purpose of assigning a probability to samples of belonging to p_{data} rather than p_g and is trained simultaneously. Accordingly, the loss function has the following expression:

$$L_{GAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z)))\right] \tag{1}$$

The discriminator perceptron tries to maximize this loss, while the generator minimizes it. This expression can be thought as the minimization of the Jensen-Shannon divergence (Arjovsky et al. 2019) between p_{data} and p_g. Figure 1 shows a diagram of the original GAN architecture.

[1] In GANs, mode drop is the name to refer to the behaviour of the generator when it underfits the data, the training process drives its weights to near 0 and greatly reduces the quality of the resulting image.

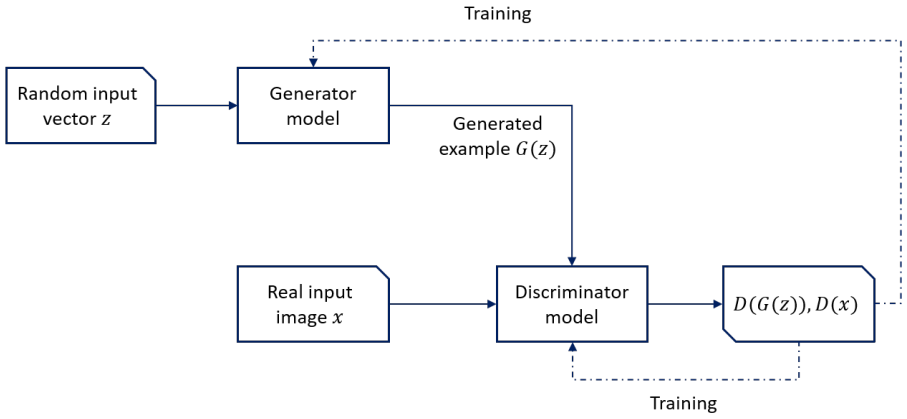


Figure 1: Original GAN architecture as described in (Goodfellow et al. 2014).

Upon the initial work on GANs, the focus was on both creating new architectures that could delve into the goodness of generative models and tweaking the loss function to reduce instability while training. A modification in the distance function led to the introduction of Wasserstein GAN (Arjovsky et al. 2017) and Wasserstein GAN with Gradient Penalty (Gulrajani et al. 2017) later. Wasserstein GANs relied on the *Earth-Mover* (EM) distance instead of Jensen-Shannon to find the stability in the system. This addition also helped to improve stability and introduced a loss metric that could be used to measure training performance because it correlated with visual quality for the first time. Nevertheless, the arbitrariness of the results made it impossible to transfer them between experiments, so the metric was limited to assess the convergence of a specific run.

As far as GAN architectures are concerned, the advances started with DCGAN (Radford et al. 2016), which expanded the possibilities of GANs to Convolutional Neural Networks (CNNs) by replacing the dense layers in the multilayer perceptrons with convolutional layers. From then on, several other architectures were proposed, such as ProGAN (Karras et al. 2018), which trains the model increasing its resolution sequentially, or SA-GAN (H. Zhang et al. 2019), which leveraged the self-attention block from (X. Wang et al. 2018) to make use of long dependencies present in images. StyleGAN (Karras et al. 2019a) and StyleGAN2 (Karras et al. 2019b) are of special relevance to this work, as they will serve as a baseline to compare the results. These two architectures propose a novel way to take control of GANs output and latent space by applying a non-linear transformation to features in the training space to map them into the "style" space before feeding it to the synthesis network.

2.2 Overfitting

There is no consensus on the notion of overfitting in generating models. For classification tasks, for instance, overfitting is produced when the model memorises the label of every sample in the training set and therefore fails in unseen data. For GANs, however, this definition is more intricate mainly because two neural networks are learning at the same time and therefore two different overfitting possibilities, depending on the side we are looking from: first, that the generator overfits means that it learns how to fool the discriminator producing one image or a cyclic sequence of images, a sort of "stamp effect". Second, the discriminator overfits if it is capable of memorising the real images and hence always assigns the highest scores to them. The generator in this case learns how to produce images very similar to those of the original dataset. When we refer to overfitting in this paper, we are talking about this second option, while mode collapse refers to the former. Mode drop, on the other hand, is produced when the GAN only covers a part of the data distribution. The work by Adlam et al. 2019 provides a good overview of the overfitting phenomena from both points of view and with a theoretical perspective.

2.3 GAN EVALUATION METRICS

The evaluation of GANs has been treated in a considerable number of papers (Salimans et al. 2016), (Heusel et al. 2017), (Olsson et al. 2018) and comparative studies (Shmelkov et al. 2018), (Borji 2019). The existence of so many works in this direction confirms how controversial and relevant this topic is. In recent times, Inception Score (IS) (Salimans et al. 2016) and Frechet Inception Distance (FID) (Heusel et al. 2017), which rely on Inception Network (Szegedy et al. 2015), have led all benchmarks on GAN evaluation. However, these two metrics have potential flaws: IS is not sensitive to intra-class diversity, FID assumes the distribution of features is gaussian, and both of them use Inception Network trained on ImageNet, so it is not clear whether these metrics could be used for classes outside the scope of this dataset or they would lead to misinterpretations (Borji 2019).

Another branch of GAN metrics research relies on using the discriminator part of the GAN to assess the performance of the generator. Generative Adversarial Metric (Im et al. 2016), which confronts two GANs by swapping their discriminators; and Tournament Win Rate and Skill Rating (Olsson et al. 2018), which is inspired in the former and proposes ways to measure the results, are two of the most known methods in the literature that could be classified in this category. These methods, nonetheless, do not provide a way of comparison between different experiments, as they use the discriminators that have been trained for the specific task, and therefore make replication of results very hard.

2.4 COMPLEX NUMBERS AND FREQUENCY DOMAIN ANALYSIS

The literature on complex theory applied to Deep Convolutional Networks is not extensive and, to the authors' knowledge, complex-valued convolutional networks are not widely used. Some works (Bruna et al. 2015), (Guberman 2016), (Worrall et al. 2017) have outlined the advantages of introducing complex filters in CNNs due to their rotation invariance, an ability that is "learnt" in real-valued CNNs by feeding them with rotated images obtained with data augmentation (Ian Goodfellow, Yoshua Bengio 2015). There are also some works on rotation invariant CNNs (Group Equivariant Convolutional Networks (Cohen et al. 2016)) that yield better results than usual CNNs when applied to medical image segmentation (Veeling et al. 2018).

However, complex numbers are generally used in frequency domain analysis rather than in spatial domain. In (X. Zhang et al. 2019), the authors describe how the up-sampling modules present in the generator of GANs generate artifacts that are barely visible as a checkboard pattern in the spatial domain but are definitely noticeable in the frequency domain (Fig. 2). For the generator to get rid of those patterns, the learned convolution kernels must behave like low-pass filters, which makes the generator incapable of learning high frequency features properly. S.-Y. Wang et al. 2019 also compare the frequency domain results of a series of GAN models, showing that some of them are more prone to present artifacts than others. Dzanic et al. (Dzanic et al. 2019) show the differences in Fourier spectrums of real and generated images by firstly reducing their dimensionality by calculating the average in circular bins of the spectrums, fitting an exponential decay function to highlight differences in the high-frequency characteristics, and later training an SVM to detect the fake dataset.

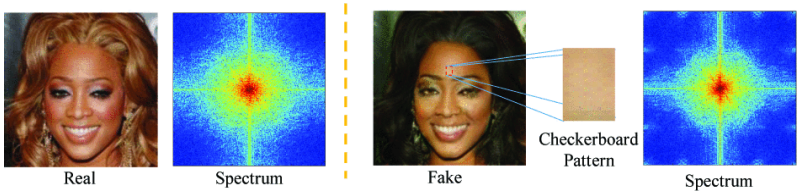


Figure 2: Comparison of the spectra of a real face image and a fake image. The detail shows the checkerboard pattern present in the spectral domain. Source: (X. Zhang et al. 2019).

3 METHODS

As it has been explained in the previous sections, for a metric to be universal, it should return the same value regardless of the orientation of the output images. This feature is particularly beneficial in medical images, in particular brain images through an axial plane, which do not possess any preferred orientation. In this section, we use the idea of dimensionality reduction through bining (Dzanic et al. 2019) to develop a rotation invariant metric using the Fourier spectrum of the dataset. We propose a method to quantify the distance between the dimensionality reduced spectrum of real and synthetic images without making use of any classification model.

3.1 REDUCTION OF DIMENSIONALITY

Let f be the value of the pixels of an image with dimensions $W \times H \times C$, defined by $f : \mathbb{R}^3 \to [0, 1]$. Constraining the values of $\mathbb{R}^3$ to (x, y, z), where x and y take continuous values in the range $[1, W]$ and $[1, H]$ respectively and z is a discrete variable that spans $\{1, C\}$. Let us denote by $\hat{f}(k_r, \theta, z)$ the Fourier transform (spectrum) of f in polar coordinates k_r, θ and z: distance to the center, angle and channel number respectively. Note that this is a 2D Fourier transform applied in the axes (x, y) and for that reason z axis does not change. We reduce the dimensionality of the Fourier spectrum summing over θ, considering it continuous and therefore define the following reduced spectrum:

$$m'(k_r, z) = a \int_0^{2\pi} |\hat{f}(k_r, \theta, z)| d\theta, \tag{2}$$

where a represents an image dependant normalization factor such that the condition $\max\{m'(k_r, z)\} = 1$ is fulfilled, thus constraining m' to the interval $[0, 1]$. Given that the number of channels is discrete, we can further reduce the dimensions by taking the L_2 norm through that axis and normalizing accordingly:

$$m(k_r) = \frac{1}{\sqrt{C}} \sqrt{\sum_{i=1}^{C} m_i'^2}, \tag{3}$$

where $m'_i \equiv m'(k_r, i)$. The idea behind the choice of L_2 norm before others (one could think that, as m' is definite positive, L_1 would be as good) lies in weighting bigger values higher. The analysis of the impact of other types of norms is left for further studies.

The magnitude in equation 3 can be computed for a group of N images, taking the average later and resulting in the Fourier spectrum of the group (denoted by $M(k_r)$ hereafter). In order to reduce the computational cost, we have decided to firstly compute the average spectrum and then calculate M right away taking its L_2 norm:

$$M'(k_r, z) = a \int_0^{2\pi} \sum_i^N |\hat{f}_i(k_r, \theta, z)| d\theta \implies M(k_r) = \frac{1}{\sqrt{C}} \sqrt{\sum_{i=1}^{C} M_i'^2} \tag{4}$$

In order to make comprehension of these steps easier, Figure 3 shows a graphical illustration of the process.

This magnitude has an error associated ($M \pm \Delta$) that stems from two sources: 1) the averaged spectrum and 2) the norm through the channels' dimension. The first error is computed as the sampling variance $\sigma^2(k_r, \theta, z)$, and is propagated to the integral. Then, the second source of error can also be propagated assuming that M'_i is statistically independent to M'_j for every $i \neq j$:

$$\Delta'(k_r, z) = \sqrt{\int_0^{2\pi} \sigma^2(k_r, \theta, z) d\theta} \implies \Delta(k_r) = \frac{1}{\sqrt{C}} \sqrt{\frac{\sum_i M_i'^2 \Delta_i'^2}{\sum_i M_i'^2}} \tag{5}$$

3.2 QUANTIFYING THE DISTANCE BETWEEN TWO CURVES

Therefore, for a group of images we have a one dimensional measurement of the spectrum, with an error associated to every point. Taking an original dataset and building another one with generated

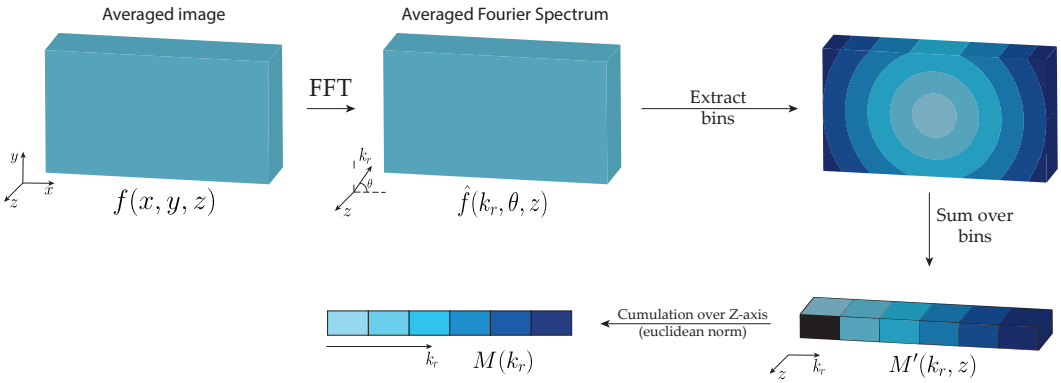


Figure 3: Graphical illustration of the methodology followed to obtain $M(k_r, \theta)$. The starting point represents the averaged image. Then, the Fourier Transform is applied to get the Fourier spectrum of this averaged image over the whole dataset. Later on, this image is separated and summed in bins and finally accumulated over channels.

images, one can compute $M_o \pm \Delta_0$ and $M_g \pm \Delta_g$ (original and generated respectively) and calculate the distance between the curves. Assuming that errors are distributed normally and that k_r takes k discrete values and therefore both curves are polygonal, it has been proved that Fréchet distance is a good measure of the similarity between them (Alt et al. 1995). Theoretically, in continuous space, given two curves $f : [a, b] \to V$ and $g : [a', b'] \to V$ in the metric space (V, d) with parametrizations α and β respectively, their Fréchet distance is defined in (Eiter et al. 1994) as

$$\delta_F(f, g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \mathrm{d}(f(\alpha(t)), g(\beta(t)))$$

When dealing with two polygonal curves P and Q, the discrete Fréchet distance between them is defined as

$$\delta_{dF}(P, Q) = \min\{\|L\| \mid L \text{ is a coupling between } P \text{ and } Q\}, \tag{6}$$

given the line segments $\sigma(P) = (u_1, ..., u_p)$ and $\sigma(Q) = (v_1, ..., v_q)$. The coupling L between P and Q is the sequence of tuples that couple the line segments preserving the order of the points in P and Q. The length $\|L\|$ of this coupling is defined by the length of the longest link using the distance defined in the metric space:

$$\|L\| = \max_{i=1,...,m} \mathrm{d}(u_{a_i}, v_{b_i})$$

3.3 Circular Spectrum Distance

In our space, P and Q correspond to the M_o and M_g curves, but every point follows a normal distribution centered in M with standard deviation Δ. Hence, we will define a metric as Circular Spectrum Distance, or CSD, as the maximum distance of the coupling of M_o and M_g using the two first momenta of the distribution at each point:

$$\text{CSD} \equiv \max_{i=1,...,k} \mathrm{d}(M_{o_i}, M_{g_i}) = \max_{i=1,...,k} \{\mathrm{d}_{euc}(\mu_{o_i}, \mu_{g_i}) + \Delta_{o_i} + \Delta_{g_i} - 2\sqrt{\Delta_{o_i} \cdot \Delta_{g_i}}\} \tag{7}$$

where $\mathrm{d}_{euc}(\mu_{o_i}, \mu_{g_i})$ represents the Euclidean distance in $\mathbb{R}$ between the first momenta of the original and generated distributions and i spans over all the possible values of k_r. Note that we are only taking the maximum of every coupling, not minimising later as expected by eq. 6. This is because we consider that the M_o and M_g curves have their nodes equally distributed, so there is only one possible path when computing discrete Fréchet Distance. We prove in Appendix B that the conditions to consider CSD a proper metric are fulfilled.

As presented in algorithm 1, the implementation of this metric to be executed during training is straightforward.

Intuitively, and according to the hypothesis that GANs act like low-pass filters, CSD should lower as training progresses because the GAN would be trying to mimic the lower frequencies of the original

dataset. However, at some point it should either remain constant or increase because the generator has acquired the optimal low-pass filter state. From that point, as training goes on, the GAN would be overfitting the original distribution.

Algorithm 1: Implementation of the Circular Spectrum Distance (CSD)

Function `mean_var` (μ, σ^2, B) **:**
  Result: M' and Δ' as defined in equations 4 and 5
  Input: $\mu \rightarrow$ Average 2D spectrum of the data
    $\sigma^2 \rightarrow$ Variance of the data
    $B \rightarrow$ Number of bins to discretize the radial dimension
  while $i < B$ **do**
    $M'_i \longleftarrow$ mean($\mu[i]$) `// Radial mean`
    $\Delta'_i \longleftarrow \sqrt{\text{sum}(\sigma^2[i])}$ `// Radial variance`
  end
  return M', Δ'
return
Function `Main`**:**
  Result: Circular Spectrum Distance as given by equation 9
  $s \longleftarrow$ spectrum(original dataset)
  $M', \Delta' \longleftarrow mean\_var(s)$
  $M_o, \Delta_0 \longleftarrow$ eqs. 4 and 5
  foreach *epoch* **do**
    $g \longleftarrow$ generator()
    $s \longleftarrow$ spectrum(g)
    $M', \Delta' \longleftarrow$ mean_var(s)
    $M_g, \Delta_g \longleftarrow$ eqs. 4 and 5
    CSD $\longleftarrow$ eq. 7
  end

4 Architecture and experimental setup

As a baseline to prove the metric proposed in the previous section, we use the recently introduced architecture StyleGAN (Karras et al. 2019a) to learn and generate MRI.

StyleGAN differs from traditional GANs like DCGAN (Radford et al. 2016) in the architecture of the generator: the latter uses a deep convolutional neural network to map the latent space to image space without additional information. Whereas StyleGAN firstly uses a fully connected neural network to map to an intermediate latent space $\mathcal{W}$ that later controls the generator using adaptive instance normalization (Huang et al. 2017). The corresponding latent space is then fed to each convolutional layer. This mapping architecture is therefore not constrained to any predefined distribution but it is learnt during the training process, which allows a more linear entanglement and a higher capacity to detect features in $\mathcal{W}$. For this reason, this first stage is responsible for the style of the final image.

There are also differences with respect to traditional GANs in the second stage of the generator, referred to as *synthesis network* in the original work: uncorrelated Gaussian noise is added to every convolutional layer with learned transformations in order to achieve a stochastic component and let the architecture produce finer details. The ablation studies do not show a major improvement in terms of FID with the addition of noise, as it does not have an effect in the high-level aspects, but the results are more vivid to the human eye.

The training architecture used in this study leverages Exponential Moving Average (EMA) to prevent cycling around the optimal solution (Yazıcı et al. 2018). The weights of the generator (θ) are updated every ten iterations according to the equation

$$\theta_{EMA}^{(t)} = \beta\theta_{EMA}^{(t-1)} + (1-\beta)\theta^{(t)} \tag{8}$$

Even though the use of this technique cannot be theoretically justified entirely, in (Yazıcı et al. 2018) the authors show that Exponential Moving Average converges to less amplitude stable limit cycles around the solution in the simpler bilinear cases. Also, they show that it improves the quality of generated images in terms of FID and IS.

4.1 Preparation of the dataset

Nuclear Magnetic Resonance images (MRI) come in a variety of modalities depending on the acquisition parameters, which affect the type of information that is captured. For this work, we have used T1 and FLAIR images:

- T1-weighted images rely on the different times of relaxation of the tissues' net magnetization vector to distinguish the components that form the body. For example, fat tissue's magnetization vector realigns quickly, while water takes more time. For this reason, fat tissue appears bright while water is darker in T1 weighted images. (Mitchell et al. 2004)

- T2-weighted images require longer inversion times than T1 because T2 rely on the time that the axial spin of the protons takes to fully arrive at their resting state, while T1 measures the time taken for the magnetic vector to return to its resting state.

- Fluid attenuation inversion recovery (FLAIR) is a special inversion recovery sequence with a long inversion time. This removes the signal from the cerebrospinal fluid in the resulting images. Brain tissue on FLAIR images appears similar to T2 weighted images with grey matter brighter than white matter but the cerebrospinal fluid is dark instead of bright. (Herlihy et al. 2001)

The dataset consists of 3D MRI images for 168 patients. Every patient has a pair of T1 and FLAIR images associated. The T1 with dimensions $256 \times 252 \times 256$ and a spacing of $(0.97, 1.00, 0.97)$mm in each axis and the FLAIR with $256 \times 112 \times 256$ and the same spacing. These images were BIAS field corrected, which is done separately for T1 and FLAIR, then co-registered and finally the T1-weighted image was resampled to FLAIR space. FLAIR and T1 images are captured in different moments of the acquisition process, potentially causing significant discrepancies that arise from involuntary movements. The co-registering process is meant to erase these discrepancies between T1 and FLAIR by moving the T1 image to FLAIR space. The objective with resampling is to find a correspondence between the voxels of T1 and FLAIR images. All of these transformations were done with 3DSlicer (*3D Slicer* 2020), (Bau et al. 2019).

Initially, the images in the dataset contain data that does not belong to the brain itself, like the skull. After some experiments with StyleGAN interpolation, we decided to use only the areas that were actually relevant for the analysis. HD-Bet brain extraction tool (Isensee et al. 2019) was used to extract the inner structures of the brain. 3D images were firstly cropped to fit the region of interest to the borders of the images and then they were resized with linear interpolation to $(128, 256, 256)$ to feed their axial planes into the neural network. Those images with no relevant information (all pixels with the same value) were removed from the dataset. As a result, every case volume is composed of over 120 images with dimensions 256×256, so more than 20,000 images take part in the final dataset. Input images have two channels corresponding to T1 and FLAIR. We chose to generate images in the axial plane because they present symmetry that is easier to learn for the generator. Figure 4 shows three of the resulting FLAIR images in the dataset.

The reasoning behind the choice of this dataset is, as it has been explained in previous sections, its underlying rotation invariance. Even though we have trained the model on aligned images, all facing the same direction, the results should not vary upon rotation of the images. Another good reason to choose a dataset composed of brain MRI is the low diversity present in the images. Compared to other datasets like CelebA or CIFAR10 (Krizhevsky 2009), where finding two similar images would be hard, the slices of the brain are very alike for two healthy subjects. This feature will help us reach one of the central conclusions of this work, as the generative model will tend to memorize the brain slices and therefore we will find generator overfitting very early into the training process.

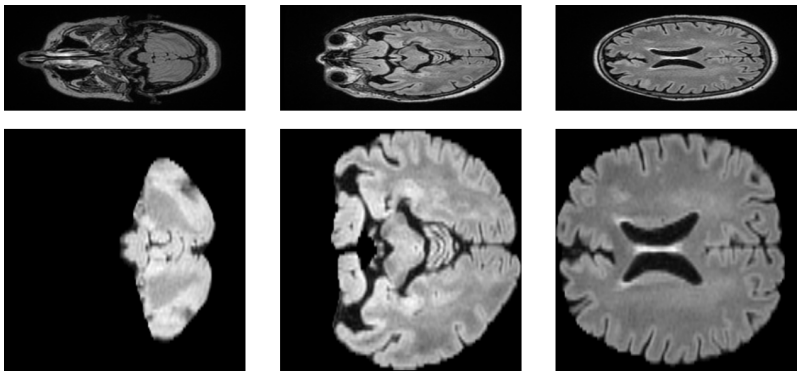


Figure 4: Detail of the steps carried out to prepare the dataset. Top row: raw data with spacing $(1, 1)$. Bottom row: data after extracting the relevant information, cropping black pixels and resizing to 256×256 using linear interpolation.

4.2 Training setup

StyleGAN with Exponential Moving Average was trained on the 20,000 images dataset, each being $256 \times 256 \times 2$. No other data augmentation techniques were applied. We used the architecture defined in Appendix C. A batch size of 8 was used and we chose a dimensionality of 512 for the input latent space. Style mixing was applied randomly with equal probability. The learning rate was chosen to be 4 times larger for the discriminator to stabilize training, as noted in (Heusel et al. 2017). We did not use any progressive growing method in the generator and we trained the architecture using WGAN loss with gradient penalty (WGAN-GP) (Gulrajani et al. 2017). No information from CSD metric was added to the loss function.

5 Results and discussion

In this section, we show the behaviour of two metrics (Frechet Inception Distance and Circular Spectrum Distance) during training. It is also shown how we implemented a quantity for overfitting that can give a rough idea of when a GAN's generator is memorising the original images and compare this vague metric to CSD. We present the results for an experiment where the original dataset is distorted to different levels of intensity to show that CSD correlates to good quality of images according to subjective evaluation. Finally, experiments with the batch size help us assess the computational cost of CSD as opposed to FID and the preferred batch size to get meaningful results for CSD. Finally, we perform a less in-depth analysis with CelebA-HQ dataset (Karras et al. 2018) where it is concluded that CSD is capable of detecting memorisation by comparing different dataset sizes. The usage of CelebA-HQ is motivated by the literature, as it provides a base scenario for comparison with well-known data.

5.1 Training behaviour

The training was performed with the configuration explained in Section 4.2. It is important to highlight that no feedback from CSD was added to the GAN loss. During training, both FID and Circular Spectrum Distance (CSD) were recorded. Figure 5 shows FID and CSD over three runs with a sample of 10,000 images each. It shows that CSD is noisier but the tendency is clear: after a period of stabilization in the first epochs, it increases and stagnates at a higher value, approximately 60% over the first stabilized value. Table 1 characterizes this increasing, showing the value of stabilization for every run before and after the ascent. It also shows the moment in Epochs when the ascending curve reaches the middle of its journey. FID, however, escalates constantly with lower slope (increases its value by 20% after 1M epochs) and with lighter variations during the whole training, thus bringing very little feedback on how the training is performing. This exact behaviour has been shown in all the runs.

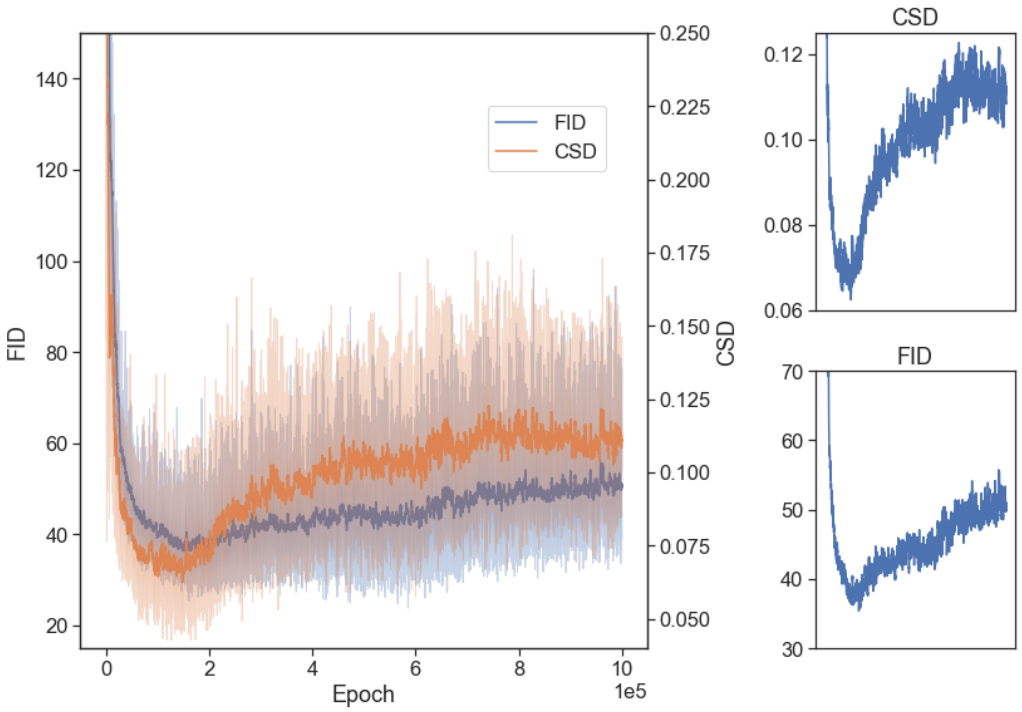


Figure 5: Left: Comparison of FID (blue) and CSD (orange) metrics performance during training. Right: Detail of one of the runs for FID and CSD.

Run	CSD before ascension	CSD after ascension	Epoch of middle point
1	0.0812	0.1169	219600
2	0.0486	0.1017	420200
3	0.0653	0.1089	304200

Table 1: Stabilized CSD values during training for each of the completed runs and epoch of the middle point of the ascension.

5.2 Overfitting metric behaviour

In order to analyze the origin of the rise of CSD after a few epochs, a simple metric of overfitting that does not rely on subjective analysis has been designed. Given a generated image, one can use the last convolutional layer of the discriminator to get its representation in a high dimensional feature space and thus be able to calculate the nearest neighbours to the generated image in the original dataset, as previously performed in (Heusel et al. 2017). After applying a *flatten* operation to the last convolutional layer (Table 3 in Appendix C), this results in a feature space of $12,288$ dimensions. Calculating nearest neighbours with brute force methods would span a considerable time with this dimensionality. Therefore, in order to reduce the time of computation, a PCA algorithm has been trained to reduce the feature space to $1,500$ principal components, explaining 95.2% of the variance. Table 4 in Appendix D shows the variance explained of the original dataset by number of variables. The usage of PCA has been explored in previous papers (Bojanowski et al. 2018) and it has been demonstrated that there are better ways of reducing the dimensionality of the feature space to retain relevant features. However, as we only want to have a rough idea of the nearest neighbours for a given image, we will use PCA for this purpose. The algorithm is capable of finding the nearest neighbours and the results are undoubtedly good (Figure 6).

Using the previous nearest neighbours algorithm we can find the minimum euclidean distance of a generated image to another in the dataset. This is reiterated in a batch of 2000 images and the

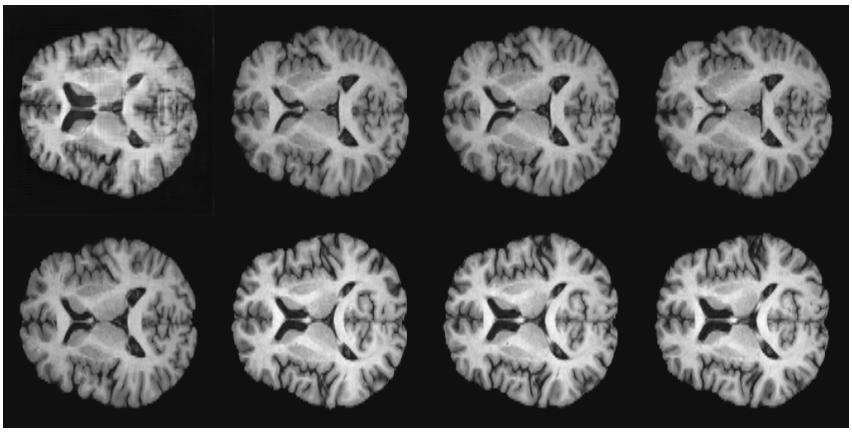


Figure 6: Nearest neighbours in the last convolutional layer feature space after applying PCA to reduce dimensionality in 88%. The synthetic image is in the top left, the other seven images belong to the dataset.

average distance of the batch yields the final value:

$$\text{Overfitting quantity} = \frac{1}{N}\sum_{i=1}^{N_g} \min_{j=1,\dots,N_o}(d_e(g_i, o_j)), \tag{9}$$

where d_e is the euclidean distance and N_g and N_o stand for the generated batch size and the original dataset size; g ($i \in [1, N_g]$) is the generated dataset and o ($j \in [1, N_o]$) is the original dataset.

This overfitting quantity has been computed every $10,000$ iterations of the batch after training, using a fully trained discriminator (1M epochs). Figure 7 compares the CSD and the overfitting quantity in the dataset during training. As can be seen, the overfitting quantity decays steadily while CSD increases because the model is learning the distribution of the original dataset and hence worsening the metric. At an early stage, the GAN is adjusting the weights to take account for the bigger details of the image, which correspond to the lower frequencies in the Fourier spectrum. At a given moment there is no more place to adjust low frequencies further, and it starts trying to copy the original dataset, thus reducing the distance even further but increasing CSD because this improvement comes from a trade-off between lower and higher frequencies. A visual inspection of the results in Figure 6 illustrates how the generator is memorising the intermediate slices of the brain: top-left image corresponds to a generated brain slice, while the rest are the nearest neighbours found for that artificial slice. As can be seen, the effect of memorisation is clear: all examples are very similar to the generated one.

Another way of grasping the level of memorisation of the generator is to look at its capacity to interpolate the latent space. Karras et al. 2019a already demonstrated that StyleGAN is capable of learning intermediate representations that are not present in the original dataset, understanding how an image transforms to another one by going through intermediate steps that make sense in image space. This is due to the inherent disentanglement of the latent space $\mathcal{W}$, given by the styling network, which makes it possible to represent features as linear transformations. In fact, since the introduction of GAN architectures capable of interpolating in latent space (which coincided with the introduction of StyleGAN), the debate of overfitting understood as generator's memorisation has practically vanished. In order to account for this effect in the generator, Figure 8 shows two samples of interpolation generated with a fully trained model (1M epochs). Odd rows contain the generated images, while even rows represent its nearest neighbour in the original dataset obtained with the embeddings of the discriminator. The presence of memorisation is crystal clear at the light of the results. For example, the third image in the third row shows a slight deformation on the right side of the brain (upper part) that is also present in an image from the dataset. These results demonstrate that in this case, the GAN has probably assigned a training image to many of the points of the latent space, and therefore it is not capable of making up new samples to travel from image A to image B.

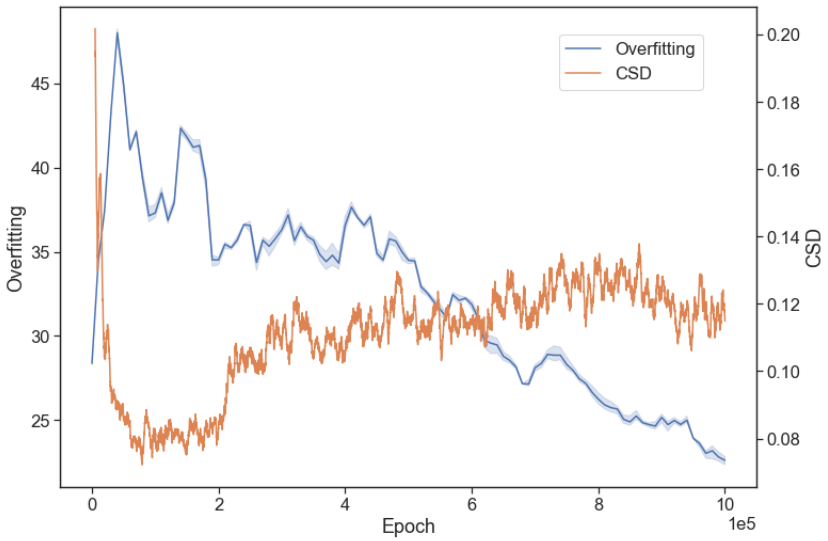


Figure 7: Comparison of the distance to the dataset calculated with equation (9) (blue) to CSD (orange)

5.3 Correlation with subjective evaluation

A metric for quality assessment in image generation tasks is only valid if it can evaluate the level at which images agree with reality (i.e. visual quality). Therefore, we carried out experiments of image distortion over the original dataset and calculated CSD for each of them, wishing to demonstrate that the metric defined as in equation 7 is meaningful under a variety of distortions. As shown in Figure 9, we blurred the dataset, added Gaussian noise, square random positioned patches, salt and pepper noise, and warped the images. In all cases, CSD responded to distortions by increasing its value considerably. This figure highlights that the metric is capable of discerning real images inside a bigger dataset.

We applied every distortion at different levels of intensity:

- Gaussian blur. We applied a Gaussian filter with a standard deviation of the Gaussian kernel from 0 to 5 in increments of 0.5
- Gaussian noise. We applied zero centred Gaussian noise, with a standard deviation ranging from 0 to 2 in increments of 0.25. Distorted images where the result of adding original images to the Gaussian noise.
- Patches. We zeroed random 25×25 pixel parts of the images. The intensity refers to the number of patches (from 0 to 12).
- Salt and pepper noise. We applied zero centred Gaussian noise, with a standard deviation ranging from 0 to 2 in increments of 0.25. Distorted images where the result of firstly multiplying original images with the Gaussian noise and then adding the outcome to original images.
- Warping. We rolled the images with a sinusoidal wave. The intensity refers to the frequency of the wave (higher values imply higher frequency and hence more distortion).

These results show in a quantitative way that CSD is useful during training: it not only can detect overfitting but also can evaluate the quality of the generated images. There is only one drop in CSD's value in the case of patches. This can be due to the distortion itself: it patches to zero random parts of the image, when many patches are added, the image transforms into a meaningless zero image, and nor CSD nor other metrics can assess their quality.

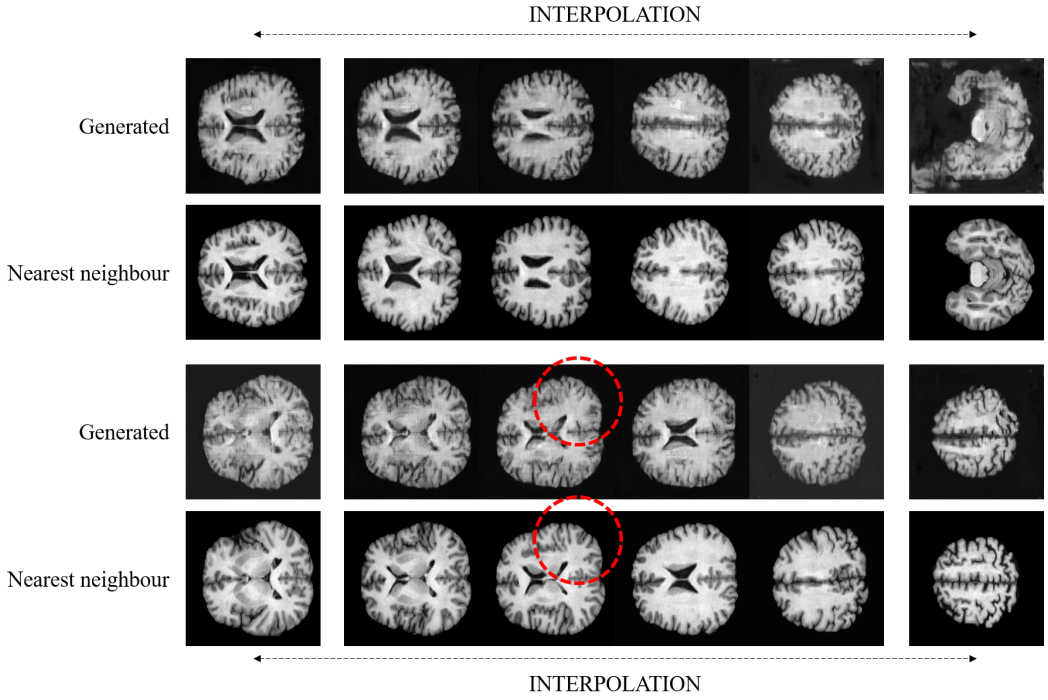


Figure 8: Effect of interpolation in two different samples. Odd rows: generated images. Even rows: nearest neighbour from the training set resulting from the embeddings of the discriminator. Red circles show two parts of the images that demonstrate the GAN has memorised the training set because they represent non average deviations of the cerebral cortex.

5.4 Metric performance analysis

The metric has been implemented in Tensorflow2 and its performance has been compared to that of FID varying the batch size. Figure 10 shows that CSD outperforms FID in terms of speed for small batch sizes. These performance measurements have been taken with the official implementation of FID in Tensorflow[2]. We have used a system with one GPU Tesla V100 with 32GB RAM and 2 Intel(R) Xeon(R) Silver 4210 CPu @ 2.20GHz.

While data for CSD rises linearly with batch size, FID is more has a logarithmic behaviour (the adjustment to this type of curve yields $R^2 = 0.87$). These measurements are very system dependant, so we can only claim that in this specific architecture, CSD is faster for small batch sizes and slowly approaches the speed of FID (potentially exceeding it) when increasing the batch size.

It is a well-known issue with FID that it needs the order of magnitude of 10k images to work properly. So many samples are needed to form the distributions of real and generated samples and be able to calculate their parameters as accurately as possible. This constraint heavily increases the time of computation, as we have previously seen in Figure 10. In order to illustrate how CSD behaves with the number of images, we have employed a fully trained generator to sample a variable number of images ranging from 1 to 5,000 and calculated CSD on that generated batch. This process is repeated five times for every batch to draw conclusions on consistent results. Figure 11 shows that CSD remains fairly constant and with a minor variance with an order of magnitude of 10^2 images. We can therefore claim that we shall be in the range where CSD is faster than FID and yet obtain meaningful results.

[2]https://github.com/bioinf-jku/TTUR

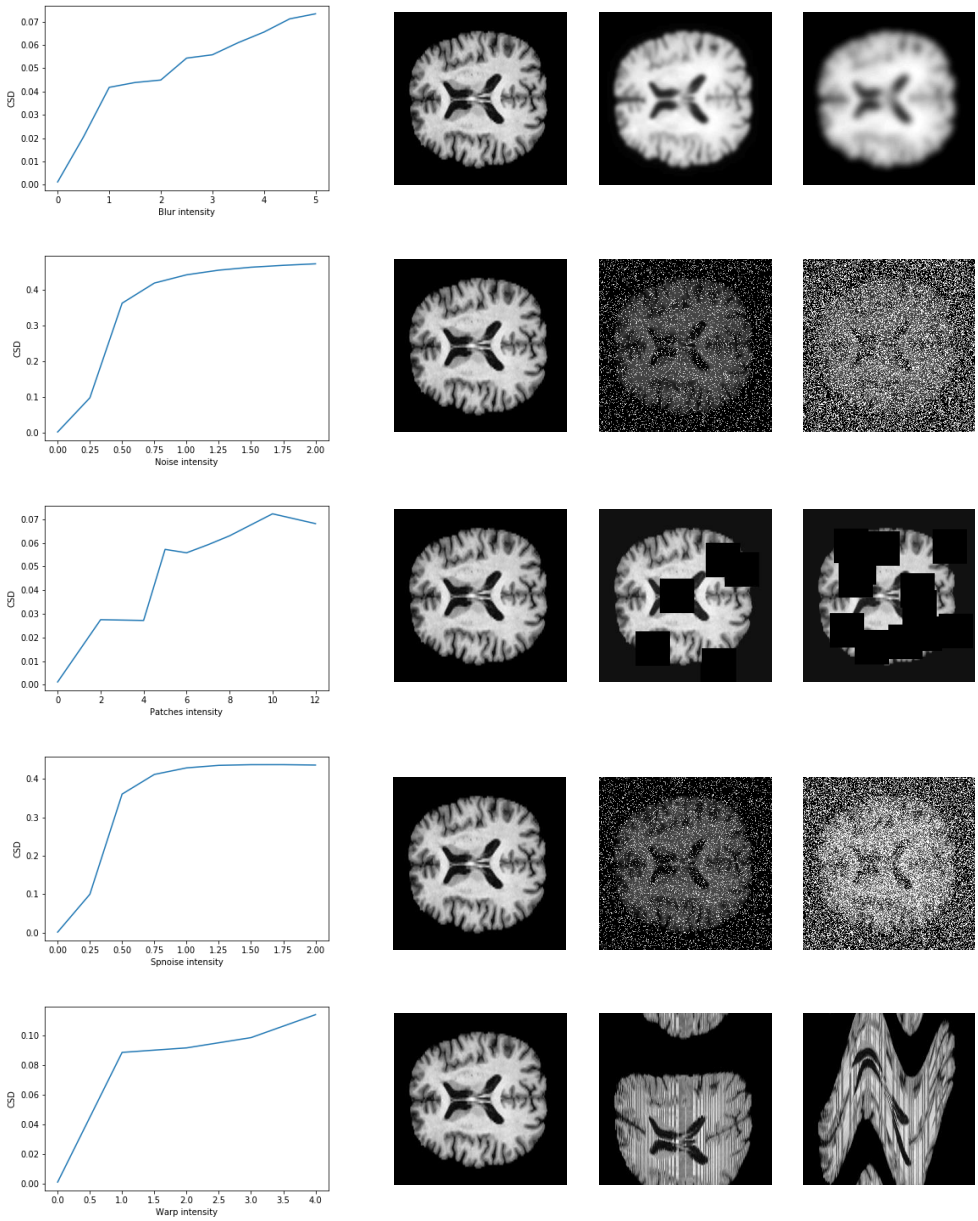


Figure 9: Correlation of CSD to image distortions. From top to bottom, distortions correspond to: blurring with a Gaussian filter, application of Gaussian noise, application of squared, random positioned patches, application of salt and pepper noise, and wrapping. **Second column** shows the real FLAIR channel from the dataset for readers' reference and **columns three and four** show the middle and highest intensity of distortion.

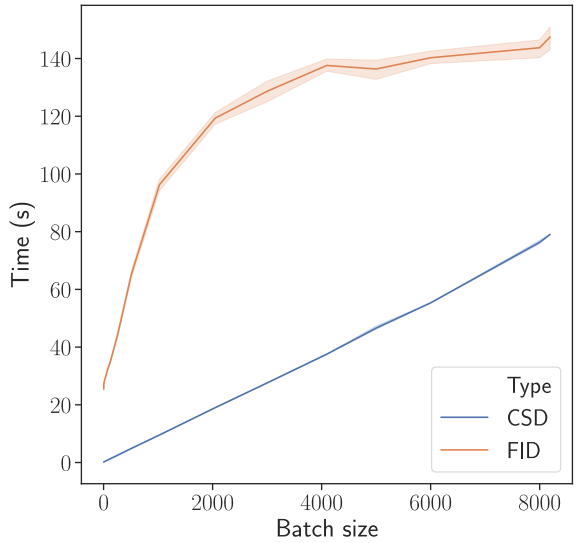


Figure 10: Comparison of FID (orange) and CSD (blue) computational cost varying the batch size and running with the same hardware configuration.

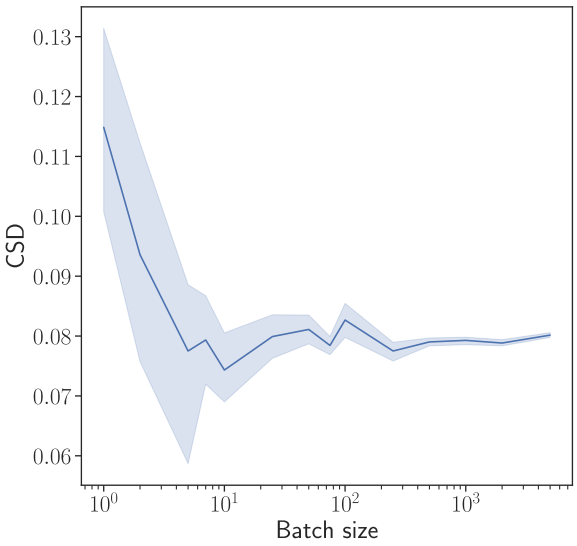


Figure 11: Correlation of CSD with the number of images. Not many images are necessary to get a decent estimate of the value of the metric.

5.5 Results in the CelebA-HQ dataset

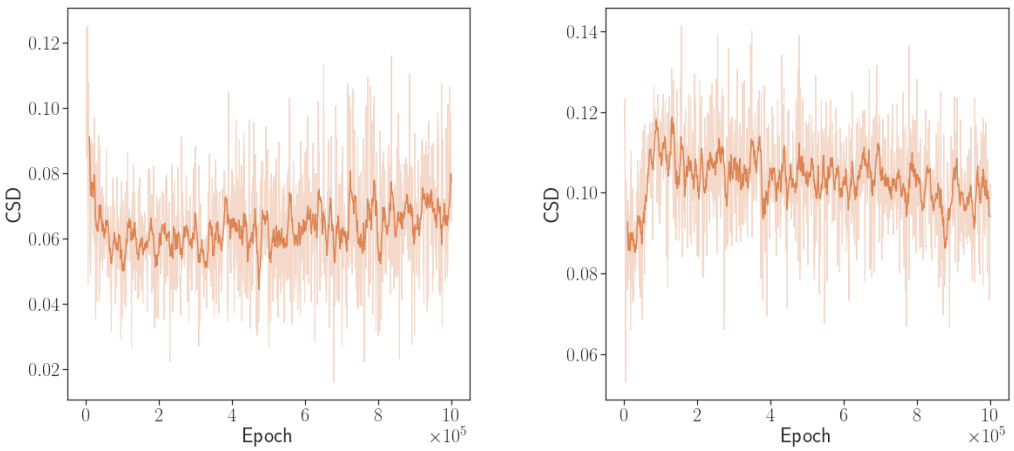
Given the special characteristics of the Brain MRI dataset we have prepared in Section 4.1, it would be at least venturous to draw conclusions on only that example. Therefore, in this section, we present the results for a different dataset (CelebA-HQ), which tipically serves as a baseline for the majority of the SOTA in image generation tasks.

CelebA-HQ (Karras et al. 2018) is a well-known dataset that consists of 30,000 images in as much as 1024x1024 resolution. It is a high resolution version of CelebA dataset (Liu et al. 2015), where the following preprocessing steps were performed: artifact removal and super-resolution to improve visual quality, Gaussian filtering, and mirror padding to generate a pleasing depth-of-field effect and finally cropping and orienting the face region to get the final image. These steps are performed for all the images in CelebA, which are later sorted favoring those images containing a broad range of frequencies in their power spectrum. This results in a more consistent dataset that does not include people "in the wild" as CelebA, but only centred images of faces.

The advantage of using this dataset with respect to the Brain dataset stems from its variety: there are no longer two very similar images and therefore the overfitting behaviour after a reasonable number of iterations should disappear.

We trained the StyleGAN with no progressive growing on the 30,000 images that compose the dataset and constraining to a resolution of 256×256, applying mirroring at random to augment the training data. The same configuration and architecture as those explained before were used: WGAN-GP loss, a batch size of 8 during the whole training and a dimensionality of 512 for the input latent space, style mixing was performed randomly with equal probability and the learning rate was fixed to four times larger for the discriminator in order to stabilize training.

During training, we measured CSD every thousand iterations (Figure 12a). In this case, the training curve does not show any stabilization after an obvious increase in value which, according to the previous analysis for the Brain dataset, would mean that the generator is not memorising the original images. This should come as no surprise. Given the results of the original paper (Karras et al. 2019a) (which we have not tried to reproduce here), the generator is capable of interpolating the latent space and defining directions that control some of the most visible characteristics of the people in the images (glasses - no glasses, hair color or gender).



(a) Full CelebA-HQ dataset

(b) 128 images chosen randomly from CelebA-HQ set

Figure 12: Recording of CSD during training with (a) the full set of CelebA-HQ and (b) only 128 images chosen randomly from this dataset. The comparison of both images show the effect of overfitting: using the whole dataset there is no clue of overfitting neither in the training progress (a) nor in the generated images (Figure 13). Constraining the dataset to 128 images and making it prone to overfitting we get the characteristic figure of overfitting in the training progress (b).

A repetition of the interpolation analysis from section 5.2 with this new dataset confirms our findings (figure 13): using the nearest neighbours algorithm defined in section 5.2 with a PCA including 3000 components and 95.5% of the variance (table 4 in the appendix D shows the amount of variance explained with respect to the number of components), we can see that the latent space is interpolated with "made up" images instead of mere reproductions of the dataset. That figure even shows that three different points in the latent space can correspond to the same image as nearest neighbour.

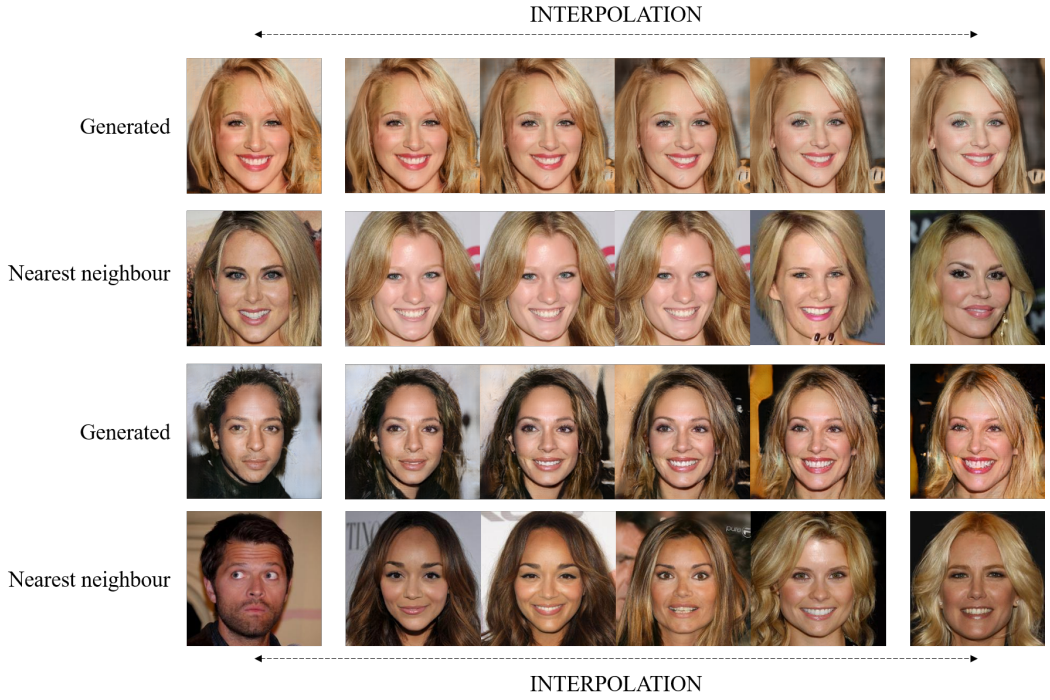


Figure 13: Effect of interpolation in two different samples using CelebA-HQ dataset. Odd rows: generated images. Even rows: nearest neighbour from the training set resulting from the embeddings of the discriminator.

It is therefore very tempting to try and "force" the generator's memorisation capabilities by reducing the number of images in the training dataset (Figure 12b). In this way, we could assess the behaviour of CSD during training and, if it works as hypothesized in Section 5.2, give a definite proof that this metric is actually detecting overfitting. With this purpose, we shuffle the dataset, take 128 images at random and let the model train for as many epochs as in the previous cases (1M), point where we could not detect any further improvement. Neither the architecture nor the model hyperparameters have undergone changes. Results will obviously lack realism due to the scarce number of training images and several artifacts stemming from the provoked overfitting.

Figure 12b shows the traning progress of StyleGAN on 128 images of CelebA-HQ. The same behaviour as in Figure 5 is presented: a brief valley followed by an increase in value and a period of everlasting stagnation. As we explained before, this is shown when there is no more place to adjust the lower frequencies of the Fourier spectrum and the generator starts memorising the dataset by forgetting about high frequencies. From the quality point of view, the comparison of Figure 12 produces a meaningful and expected conclusion, as the average value of the metric is always higher in the model that should not be able to generate realistic images.

The exploration in Figure 14 of the latent space that has been learned with only 128 images is very insightful: the model has been absolutely incapable of creating a pleasuring interpolation and has repeated representations throughout. Also, in some cases it has tried to memorise and print images already in the dataset (first image, upper left, in Figure 14).

5.6 Effect of splitting the dataset into training and testing

The reader would now wonder why we are not using the widely spread technique of separating a test dataset and checking the performance of CSD during training on both training and test sets. Intuition tells us that if the generator is memorising the training set, the metric applied to the test set should go down steadily. However, this separation is not of much value in this case because the metric is using the average spectrum to calculate the distance in each step, and this does not change much between

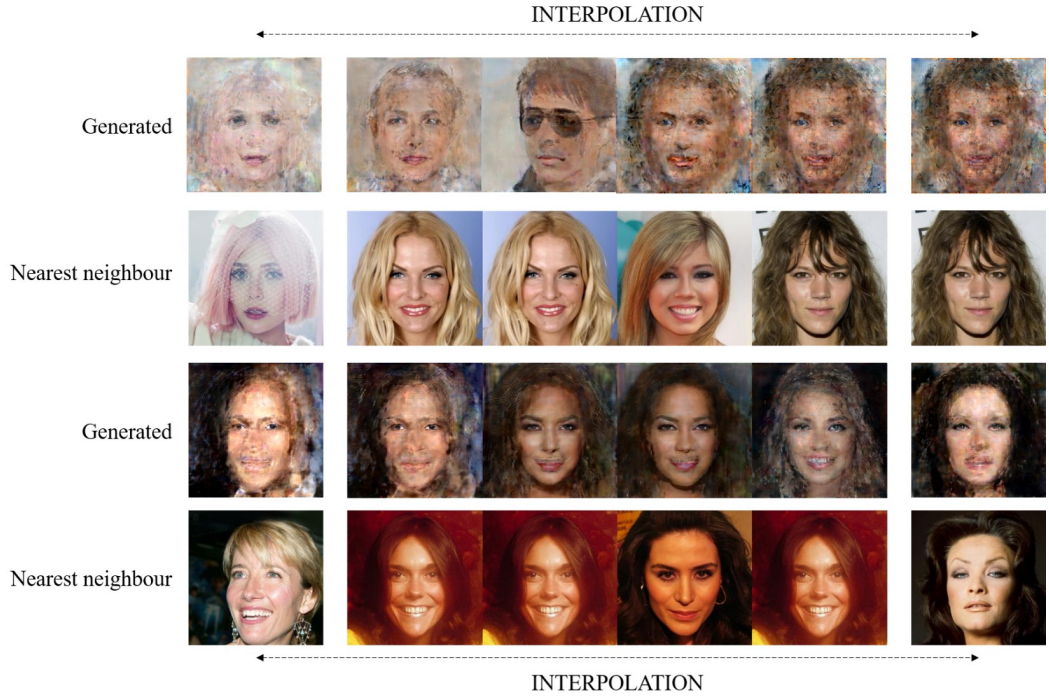


Figure 14: Effect of interpolation in two different samples using only 128 images choosen at random from CelebA-HQ dataset. Odd rows: generated images. Even rows: nearest neighbour from the training set resulting from the embeddings of the discriminator.

images. Figure 15 shows the reduced Fourier spectrum for two different images. Both curves are very similar even though the images are two separated slices in the brain.

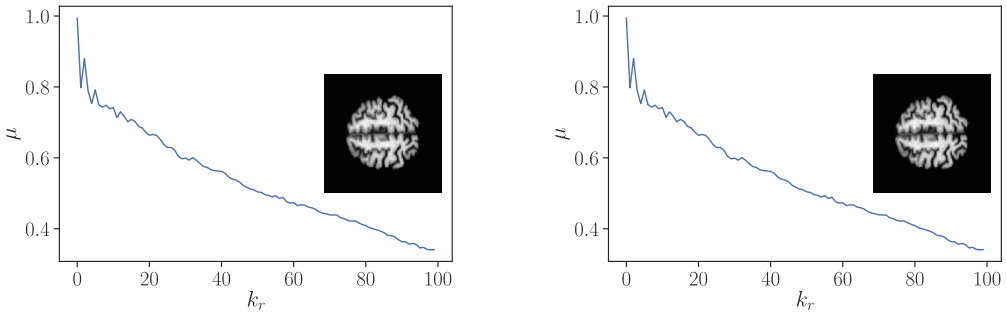


Figure 15: Reduced Fourier spectrum obtained with equation 4 for two separated slices in the brain showing that the spectrum is very similar in a dataset even when the images are different.

6 Conclusions and future work

We have introduced CSD, a metric based on the distance of original and synthetic Fourier spectra that can be used to assess the performance of the generator of GANs and that does not depend on pretrained models to operate. The hypothesis that lies under the functioning of this metric is that GANs' generators tend to work as low-pass filters and hence generated images present a high deviation from original images in high frequencies. We have presented sound evidence that the metric is valuable during training not only because it correlates with subjective evaluation but also because it draws a particular figure that is related to GAN memorisation.

To the authors' knowledge, there are no other works that delve into the usage of Fourier Spectrum to create metrics valuable for GANs. Also, there is no other metric that can detect generator memorisation and quality during training. For these reasons, we think that with this master's thesis we have expanded the limits of the current state of the art, introducing new ways of assessing GAN behaviour and actually developing a metric that has proved the expected performance under a number of different datasets.

This work opens a new horizon for deep learning in tasks of image processing because it makes use of complex analysis to account for differences in the generated images. Usage of Fourier spectra is residual in the literature in generative deep learning, limiting itself to subjective analyses and architecture comparisons. With this master's thesis, we wanted to demonstrate that even the easiest usages of this tool can be beneficial and bring impressive results. The in-depth study of complex valued networks might be profitable especially in classification tasks because of their rotation invariancy. Moreover, as future work, it could be worthwhile to experiment with different norms and analyse the metric behaviour. Also, the incorporation of this metric as a new regularization term could lead to even more undetectable fake images.

REFERENCES

3D Slicer (2020). URL: https://www.slicer.org/ (visited on 09/17/2020).

Adlam, Ben, Charles Weill, and Amol Kapoor (2019). "Investigating Under and Overfitting in Wasserstein Generative Adversarial Networks". In: arXiv: 1910.14137v1.

Alt, Helmut and Michael Godau (1995). "Computing the Frêchet Distance between two poligonal curves". In: *International Journal of Computational Geometry & Applications*. ISSN: 0218-1959. DOI: 10.1142/s0218195995000064.

Arjovsky, Martin and Léon Bottou (2019). "Towards principled methods for training generative adversarial networks". In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. arXiv: 1701.04862.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). *Wasserstein GAN*. Tech. rep. arXiv: 1701.07875v3.

Bau, David et al. (Oct. 2019). "Seeing What a GAN Cannot Generate". In: *Proceedings of the IEEE International Conference on Computer Vision*. ISSN: 15505499. DOI: 10.1109/ICCV.2019.00460. arXiv: 1910.11626.

Bojanowski, Piotr et al. (2018). "Optimizing the latent space of generative networks". In: *35th International Conference on Machine Learning, ICML 2018*. ISBN: 9781510867963. arXiv: 1707.05776.

Borji, Ali (2019). "Pros and cons of GAN evaluation measures". In: *Computer Vision and Image Understanding*. ISSN: 1090235X. DOI: 10.1016/j.cviu.2018.10.009. arXiv: 1802.03446.

Bruna, Joan et al. (2015). *A Mathematical Motivation for Complex-valued Convolutional Networks*. Tech. rep. arXiv: 1503.03438v3.

Cohen, Taco S. and Max Welling (Feb. 2016). "Group Equivariant Convolutional Networks". In: *33rd International Conference on Machine Learning, ICML 2016* 6, pp. 4375–4386. arXiv: 1602.07576.

Dowson, D. C. and B. V. Landau (1982). "The Fréchet distance between multivariate normal distributions". In: *Journal of Multivariate Analysis* 12.3, pp. 450–455. ISSN: 10957243. DOI: 10.1016/0047-259X(82)90077-X.

Dzanic, Tarik and Freddie Witherden (Nov. 2019). "Fourier Spectrum Discrepancies in Deep Network Generated Images". In: arXiv: 1911.06465.

Eiter, Thomas and Heikki Mannila (1994). *Computing Discrete Fréchet Distance*. Tech. rep.

Goodfellow, Ian J. et al. (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems*. DOI: 10.3156/jsoft.29.5_177_2. arXiv: 1406.2661.

Guberman, Nitzan (Feb. 2016). "On Complex Valued Convolutional Neural Networks". In: arXiv: 1602.09046.

Gulrajani, Ishaan et al. (2017). *Improved Training of Wasserstein GANs*. Tech. rep. arXiv: 1704.00028v3.

Herlihy, Amy H. et al. (2001). "FLAIR imaging using nonselective inversion pulses combined with slice excitation order cycling and k-space reordering to reduce flow artifacts". In: *Magnetic Resonance in Medicine*. ISSN: 07403194. DOI: 10.1002/mrm.1198.

Heusel, Martin et al. (June 2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: arXiv: 1706.08500.

Huang, Xun and Serge Belongie (2017). "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization". In: *Proceedings of the IEEE International Conference on Computer Vision*. ISBN: 9781538610329. DOI: 10.1109/ICCV.2017.167. arXiv: 1703.06868.

Ian Goodfellow, Yoshua Bengio, Aaron Courville (2015). "Deep Learning Book". In: *Deep Learning*. ISSN: 1437-7780. DOI: 10.1016/B978-0-12-391420-0.09987-X. arXiv: arXiv:1011.1669v3.

Im, Daniel Jiwoong et al. (2016). *Generating images with recurrent adversarial networks*. Tech. rep. arXiv: 1602.05110v5.

Isensee, Fabian et al. (2019). "Automated brain extraction of multisequence MRI using artificial neural networks". In: *Human Brain Mapping*. ISSN: 10970193. DOI: 10.1002/hbm.24750. arXiv: 1901.11341.

Karras, Tero and Samuli Laine (2019a). *A Style-Based Generator Architecture for Generative Adversarial Networks*. Tech. rep. arXiv: 1812.04948v3.

Karras, Tero et al. (2018). *Progressive growing of GANs for improved quality, stability, and variation*. Tech. rep. arXiv: 1710.10196v3.

Karras, Tero et al. (2019b). *Analyzing and Improving the Image Quality of StyleGAN*. Tech. rep. arXiv: 1912.04958v1.

Krizhevsky, Alex (2009). "Learning Multiple Layers of Features from Tiny Images". In: ... *Science Department, University of Toronto, Tech.* ... ISSN: 1098-6596. DOI: 10.1.1.222.9220. arXiv: arXiv:1011.1669v3.

Liu, Ziwei et al. (2015). "Deep learning face attributes in the wild". In: *Proceedings of the IEEE International Conference on Computer Vision*. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.425. arXiv: 1411.7766.

Mitchell, D G and M Cohen (2004). *MRI Principles*. Saunders. ISBN: 9780721600246.

Olsson, Catherine et al. (2018). *Skill Rating for Generative Models*. Tech. rep. arXiv: 1808.04888v1.

Radford, Alec, Luke Metz, and Soumith Chintala (2016). "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. arXiv: 1511.06434.

Salimans, Tim et al. (June 2016). "Improved Techniques for Training GANs". In: arXiv: 1606.03498.

Shao, Grace (Oct. 2019). *Deepfakes could be a big problem for the 2020 election*. URL: https://www.cnbc.com/2019/10/15/deepfakes-could-be-problem-for-the-2020-election.html (visited on 08/24/2020).

Shmelkov, Konstantin, Cordelia Schmid, and Karteek Alahari (July 2018). "How good is my GAN?" In: arXiv: 1807.09499.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298594. arXiv: 1409.4842.

Toews, Rob (May 2020). *Deepfakes Are Going To Wreak Havoc On Society. We Are Not Prepared.* URL: https://www.forbes.com/sites/robtoews/2020/05/25/deepfakes-are-going-to-wreak-havoc-on-society-we-are-not-prepared/ (visited on 08/24/2020).

Veeling, Bastiaan S. et al. (June 2018). "Rotation Equivariant CNNs for Digital Pathology". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11071 LNCS, pp. 210–218. arXiv: 1806.03962.

Wang, Sheng-Yu et al. (Dec. 2019). "CNN-generated images are surprisingly easy to spot... for now". In: arXiv: 1912.11035.

Wang, Xiaolong et al. (2018). "Non-local Neural Networks". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ISBN: 9781538664209. DOI: 10.1109/CVPR.2018.00813. arXiv: 1711.07971.

Worrall, Daniel E et al. (2017). *Harmonic Networks: Deep Translation and Rotation Equivariance*. Tech. rep. arXiv: 1612.04642v2.

Yazıcı, Yasin et al. (June 2018). "The Unusual Effectiveness of Averaging in GAN Training". In: *7th International Conference on Learning Representations, ICLR 2019*. arXiv: 1806.04498.

Zhang, Han et al. (2019). *Self-Attention Generative Adversarial Networks*. Tech. rep. arXiv: 1805.08318v2.

Zhang, Xu, Svebor Karaman, and Shih-Fu Chang (July 2019). "Detecting and Simulating Artifacts in GAN Fake Images". In: DOI: 10.1109/wifs47025.2019.9035107. arXiv: 1907.06515.

APPENDICES

A BRIEF DISCUSSION OF THE SOCIAL AND ETHICAL IMPLICATIONS

The social and ethical implications of the development of a metric for GANs have been briefly discussed previously in Section 1. In this appendix we will explore them a bit further.

GANs, just like any other invention, do not serve a good or bad purpose, it is the user who decides what to do with it. In this work we have presented a metric capable of discerning real and synthetic images and capable of detecting overfitting in terms of memorisation in GANs. Therefore, the social implications of the present work are blurred. On the one hand, GANs are an effective way of expanding datasets with synthetic images and reducing (for instance) racial bias in people datasets. On the other hand, a growing number of articles are being published concerning the effect that Deepfakes could have in the life of many people. It is increasingly easier for the general public to put words in other people's mouth and post them to social media, with devastating consequences to the affected persons. In the first case, a metric like CSD would just assess on the performance during training. In the second case, the metric could be adopted by social media platforms to disallow the publication of potentially offensive videos and images based on Deepfakes, protecting their users from unethical uses of artificial intelligence.

B FULFILLMENT OF THE CONDITIONS TO CONSIDER CSD A PROPER DISTANCE

Let A and B be two polygonal curves with n equally distributed endpoints $A = (a_1, ..., a_n)$ and $B = (b_1, ..., b_n)$, where every endpoint represents a normal univariate distribution with momenta μ_{a_i} and σ_{a_i}. Let $d(a_i, b_i)$ be the Fréchet distance between the normal univariate distributions that lie in endpoints a_i and b_i:

$$d(a_i, b_i) = d_{euc}(\mu_{a_i}, \mu_{b_i}) + \sigma_{a_i} + \sigma_{b_i} - 2\sqrt{\sigma_{a_i}\sigma_{b_i}} \tag{10}$$

We define CSD as a simplification of the discrete Fréchet distance coming from the fact that endpoints are equally distributed. Therefore, we can get rid of the minimum. Applied to A and B curves:

$$\text{CSD}(A, B) \equiv \max_{i=1,...,k} \text{d}(a_i, b_i) \tag{11}$$

Equation 11 defines a metric on the space of A and B.

Proof:

Firstly, let us note that eq. 10 defines a metric as proved in (Dowson et al. 1982). We will make use of its properties in this demonstration.

1. $\text{CSD}(A, B) = \text{CSD}(B, A)$. Because of the properties of the Fréchet distance between multivariate, normal distributions:
$$\text{CSD}(A, B) = \max_i \text{d}(a_i, b_i) = \max_i \text{d}(b_i, a_i) = \text{CSD}(B, A)$$
2. $\text{CSD}(A, B) \geq 0$ because $\text{d}(a_i, b_i) \geq 0$, so $\max_i \text{d}(a_i, b_i) \geq 0$.
If $\text{CSD}(A, B) = 0$ then $A = B$. $\max_i \text{d}(a_i, b_i) = 0$ implies $d(a_i, b_i) = 0\ \forall i$. Using the previously cited properties, $a_i = b_i\ \forall i$ and therefore $A = B$.
If $A = B$ then $\text{CSD}(A, B) = 0$. $\text{CSD}(A, A) = \max_i \text{d}(a_i, a_i) = 0$.
3. We take three curves A, B and C and show that triangle inequality is fulfilled i.e. $\text{CSD}(A, B) \leq \text{CSD}(A, C) + \text{CSD}(C, B)$.
From the definition, triangle inequality holds for $d(a_i, b_i)$: $d(a_i, b_i) \leq d(a_i, c_i) + d(c_i, b_i)$. Taking the maximum on both sides yields:
$$\max_i d(a_i, b_i) \leq \max_i \{d(a_i, c_i) + d(c_i, b_i)\} \leq \max_i d(a_i, c_i) + \max_i d(c_i, b_i)$$
because both $d(a_i, c_i)$ and $d(b_i, c_i)$ are positive. Therefore,
$\text{CSD}(A, B) = \max_i d(a_i, b_i) \leq \max_i d(a_i, c_i) + \max_i d(c_i, b_i) = \text{CSD}(A, C) + \text{CSD}(C, B)$
and triangle inequality also holds for CSD and it defines a metric.

C NETWORKS ARCHITECTURE

Style	Act.	Output shape	Params
Latent vector	-	512	-
Dense	LReLU	512	263k
Dense	LReLU	512	263k
Dense	LReLU	512	263k
Dense	LReLU	512	263k
Total trainable parameters			1.1M

Synthesis	Act.	Output shape	Params
Latent vector	-	512	-
Dense	ReLU	3072	396k
Reshape	-	$4 \times 4 \times 192$	-
Style block 1	LReLU	$4 \times 4 \times 768$	2.1M
Style block 2	LReLU	$8 \times 8 \times 384$	3.0M
Style block 2	LReLU	$16 \times 16 \times 288$	1.3M
Style block 2	LReLU	$32 \times 32 \times 192$	695k
Style block 2	LReLU	$64 \times 64 \times 144$	397k
Style block 2	LReLU	$128 \times 128 \times 96$	223k
Style block 2	LReLU	$256 \times 256 \times 48$	91k
Conv 1×1	-	$256 \times 256 \times 2$	98k
Total trainable parameters			8.2M

Table 2: Generator networks architectures. Left: styling network that maps from $\mathcal{Z}$ to $\mathcal{W}$. Right: synthesis network architecture with style blocks as in (Karras et al. 2019a)

Discriminator	Activation	Output shape	Params
Input image	-	$256 \times 256 \times 2$	-
Conv 3×3	LReLU	$256 \times 256 \times 48$	912
Average pooling	-	$128 \times 128 \times 48$	-
Conv 3×3	LReLU	$128 \times 128 \times 96$	42k
Average pooling	-	$64 \times 64 \times 96$	-
Conv 3×3	LReLU	$64 \times 64 \times 144$	125k
Average pooling	-	$32 \times 32 \times 144$	-
Conv 3×3	LReLU	$32 \times 32 \times 192$	249k
Average pooling	-	$16 \times 16 \times 192$	-
Conv 3×3	LReLU	$16 \times 16 \times 288$	498k
Average pooling	-	$8 \times 8 \times 288$	-
Conv 3×3	LReLU	$8 \times 8 \times 384$	996k
Average pooling	-	$4 \times 4 \times 384$	-
Conv 3×3	LReLU	$4 \times 4 \times 768$	2.7M
Flatten	-	12288	-
Dense	LReLU	768	9.4M
Dense	-	1	769
Total trainable parameters			1.1M

Table 3: Discriminator network architecture

D EXPLAINED VARIANCE

When running a PCA algorithm it is of crucial importance to know how much variance of the dataset is being explained with respect to the number of components. We have used this algorithm for the two datasets considered: MRI and CelebA-HQ, and the following table shows the results and the reason to choose certain number of components in each case.

# components	MRI dataset	CelebA-HQ
500	86.8	80.3
1000	92.6	87.2
1500	95.2	90.6
2000	96.7	92.8
2500	97.7	94.3
3000	98.4	95.5
3500	98.8	96.4
4000	99.1	97.1
4500	99.4	-
5000	99.5	-

Table 4: Percentage of explained variance of the MRI brain dataset and CelebA-HQ applying a PCA algorithm with the number of components pointed out in the first column.

E SAMPLES GENERATED BY THE STYLE-GAN

The two following figures show the samples generated by the Style-GAN. For the brain samples, Figure 16 shows the FLAIR images and Figure 17 shows their corresponding T1-weighted images. One can see that the network has learnt the correspondence in shape between T1 and FLAIR.

The two following show the generated images using the whole CelebA-HQ dataset (Figure 18) and only 128 images from the said dataset (Figure 19).

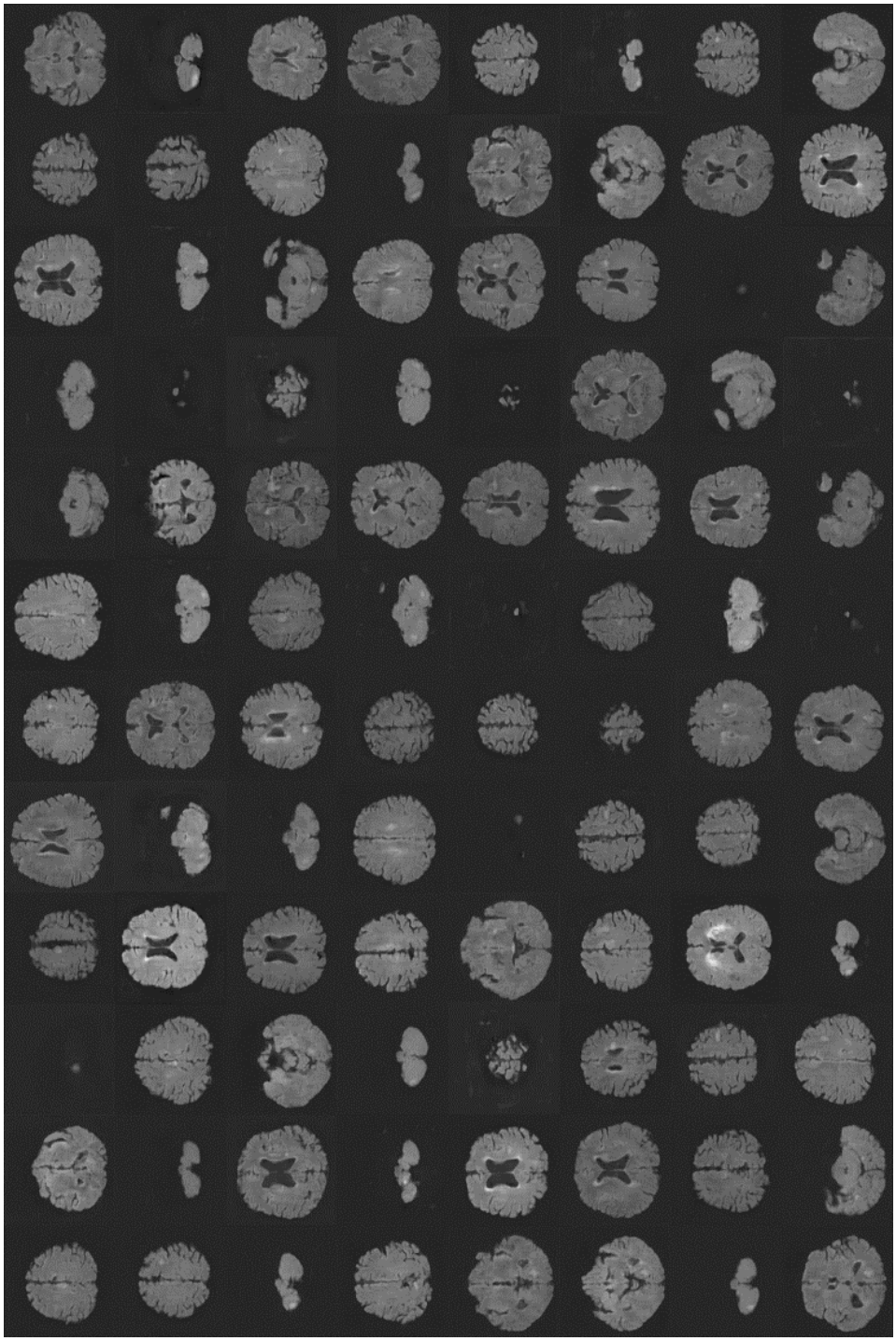


Figure 16: Randomly generated FLAIR images using Style-GAN and the dataset from Section 4.1.

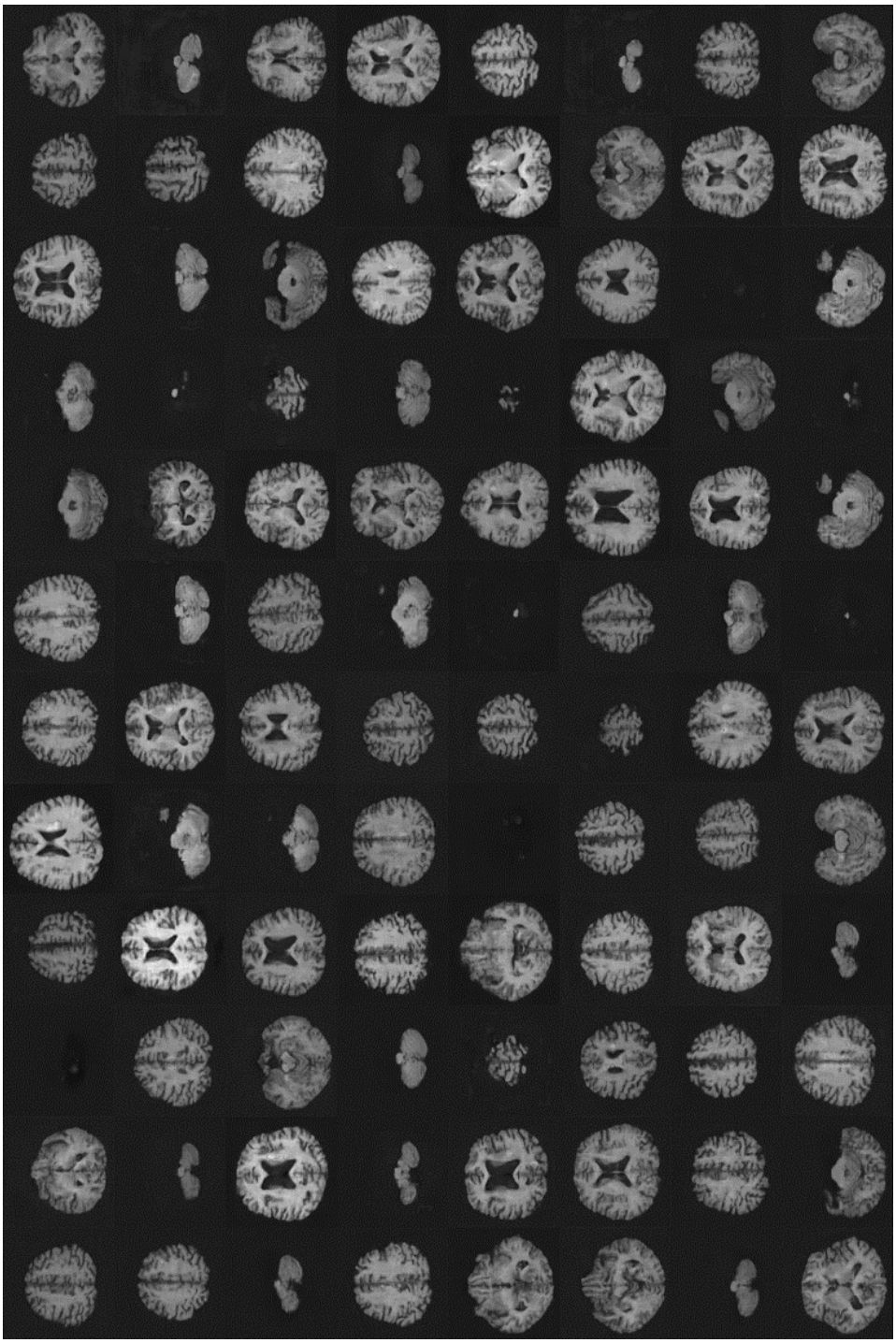


Figure 17: Randomly generated T1-weighted images using Style-GAN and the dataset from Section 4.1.

Figure 18: Randomly generated CelebA-HQ images using the whole dataset

Figure 19: Randomly generated CelebA-HQ images using only 128 images for training. Memorisation is evident.