Instituto Tecnológico
de Buenos Aires

Use of Generative Adversarial Networks for the creation and manipulation of facial images in the context of studying false memories and its effects on wrongful conviction cases

Implementation of StyleGAN's generative image modeling and style mixing properties to design an interface for experimentation purposes.

Jimena Lozano

Maite Herrán

Supervisor: Rodrigo Ramele

Final Project

Computer Engineering Degree

Abstract

El Laboratorio de Sueño y Memoria from Instituto Tecnológico de Buenos Aires (ITBA) studies the formation of false memories, and how these can be reduced or modified, and is in collaboration with the Innocence Project to investigate how these can lead to errors in convictions. From this research, the need arises to carry out experiments with human faces that are similar to each other, and how this similarity can result in the formation of false memories.

In this project, we investigate a field of Artificial Intelligence (AI), Deep Learning, which can provide us with a solution to the generation of artificial faces. In particular, we implement a face generation model using a Generative Adversarial Network (GAN), with the aim of generating faces as realistic as possible, so that a human cannot distinguish them from real faces. StyleGAN, a particular implementation of the GAN network, was the chosen architecture, because in addition to producing images with high resolution quality, it presents a model that allows navigation of the latent space and the synthesis of faces, using style mixing properties. Finally, an application called FG-Style was developed and installed on a GPU-based server at ITBA so that the laboratory can have control over the face generation model, and over the generation of faces similar to a selected one, using StyleGAN's style mixing properties to have a grip over the change of specific features of the generated faces.

Keywords – StyleGAN, GAN, Generative Image Modeling, Innocence Project, False Memories, *Laboratorio de Sueño y Memoria*, ITBA

Contents

1 Introduction

It is impossible to determine how many people have been falsely convicted throughout history. In 2011, a study which attempted to estimate the number of innocent people in prison was conducted. The conservative statistic estimated twenty thousand people were wrongfully convicted since 1980 [36].

False confessions, poor defense, bad use of forensic science and eyewitness misidentification are some of the causes why people might go to jail without having perpetrated a crime [36]. More and more institutions are agreeing that accuracy and confidence in eyewitness criminal identifications in lineups isn't reliable. Understanding how memory works when remembering faces is of uttermost importance in order to develop fair, clean and valid protocols for lineup identifications [49].

Innocence Project is a network of organizations which advocate for these unfair causes around the world. *Innocence Project Argentina* (the argentinian based organization) has paired up with ITBA's *Laboratorio de Sueño y Memoria* to deepen the knowledge on how false memories can make eyewitnesses incriminate an innocent person.

One of the experiments the Lab is currently conducting is to investigate the reliability in eyewitness criminal identifications in lineups. Subjects participate in a simulated crime and are later shown different lineups where the real criminal might be present or not. In some of the lineups, the criminal is replaced with a person with a face with similar attributes. The idea of the experiment is to see **who** the participants point out as the criminal and **which** are the attributes that the innocent person has in common with the perpetrator that make the participants mislead to choosing the wrong person. In order to make this experiment, the Lab needs to have access to faces which have similar specific attributes they want to study. Finding these faces poses technical challenges.

The current work's objective consists of the exploration of StyleGAN2, a Generative Adversarial Network which is used for the generation of images (mainly faces) in order to find different ways in which the Lab could make use of this tool to create different experiments for their studies. Having access to random faces and being able to have some kind of manipulation over them will help them in creating experiments like the one mentioned above, or even new ones to study the relationship between memory and face recognition. The idea is to also provide the Lab with an easy to use tool called **FG-Style** where they can

1. Generate artificial, realistic and high-quality facial images.

2. Obtain similar but not equal facial images.

3. Manipulate facial features of a generated image.

It is worth noting that an underlying aim of the work is to conduct the exploration of StyleGAN2 with prior knowledge in memory and false memories in order to have a closer insight into what the Lab needs or might need in the future.

As mentioned above, the chosen tool to conduct the investigation is the GAN StyleGAN2. GANs are a type of deep learning algorithms called the Generative Adversarial Networks which were introduced by Ian Goodfellow in 2014. These networks have created great expectations among the scientific community because they allow the generation of new synthetic data that is indistinguishable from real data. In 2018, NVIDIA published the StyleGAN paper “A Style-Based Architecture for GANs” where an alternative generator architecture is used which allows different levels of control of the generated images. StyleGAN2 is built on StyleGAN and has several improvements like solving errors that appeared in the first version such as symmetry of faces or the position of teeth relative to the angle of the image.

Readers of the present work will first find in Section 2 a brief introduction to human’s memory and the formation of false memories in order to understand the neuro-scientific angle of the study. Next, in Section 3, both *El Laboratorio de Sueño y Memoria* and *Innocence Project* are presented together with the link they have to help in the cause for wrongful convictions. In Section 4, a more theoretical approach to Deep Learning, Machine Learning and GANs is presented to the reader as means to understand the subjacent technologies where **FG-Style** builds upon. Furthermore, in Section 5 the whole tangible work for the face generation engine is described in depth: StyleGAN2, the GPU settings required to conduct the investigation, the conceptual vectors and all of the work done to generate and manipulate the faces. In Section 6 readers will find explained the final solution design and the back end application together with the front end application that will be of use to the Lab. Then Section 7 explores the results achieved while Section

8 alludes to the demonstration of the work done to the director of the Lab, Dr. Cecilia Forcato. Finally, the work wraps up with future implementations and the final conclusion of the whole work. The Appendix comments on an open online talk which was organized by the NGO *Women in Data* and whose speakers are the authors of this paper, and also comments on a collaboration done with UBA and CONICET's *Instituto de Ciencias de la Computación* (ICC) researcher PhD Pablo Negri.

2 Memories and False Memories

2.1 Memories

Before we study false memories, we should first ask ourselves a series of questions. What if one day we wake up with no memory? What are memories? What does learning mean?

The textbook Principles of Neural Science defines learning as "a change in behavior that results from acquiring knowledge about the world", and defines memory as "the process by which that knowledge is encoded, stored, and later retrieved” [20]. Memory is encoded in neural circuits, through changes in the properties of neurons and in the connections between them thus forming an internal representation of information.

When a memory is formed, it is not formed straight away, otherwise we would retain every piece of information we are exposed to: we could store in our memory all of the titles and authors of our local library’s books just by taking a tour through its corridors and glancing at the titles. Memory formation has a temporary dynamic and the steps of memory formation are acquisition, consolidation, retrieval and reconsolidation [11].

1. **Memory acquisition** is the moment at which a memory begins to form and it consists in the encoding of the information. This memory is weak, labile and its formation can be interrupted.

2. **Memory consolidation** consists of the process when the memory goes from a labile state to a stable state. It happens over a period of time. If, for instance, in that period of time a human is given an inibidor of protein synthesis, it will impair the consolidation because it depends on protein synthesis and gene expression [19] [34]. Some synonyms of memory consolidation are stabilization and storage.

3. **Memory retrieval** is the recovery of the information stored. It lays bare if the memory is stored or not. However, the fact of not being able to evoke information does not mean that a memory is not stored. One could have a memory and not be able to evoke it.

4. **Memory reconsolidation** occurs when an old memory is reopened and its content is modified. The re-stabilization of the content is called reconsolidation and it

> is dependent on gene expression and protein synthesis in the circuit involved in memory. Some time ago, it was thought that a memory, once formed, remained intact throughout time and only fell into oblivion. Subsequent studies have shown that if one exposes a key that has been linked to that initial learning process, memory can be reactivated, return to a state of lability (a vulnerable state) and in order to persist in time, it will go through the process of reconsolidation [13].

Reconsolidation is the phase of memory formation which we are going to study in more detail. It can occur that an animal or human that has stored, attached and consolidated an old memory can be re-exposed to a key that was present at the time of learning that old memory, and the memory can be weakened, it can be opened. Having said this, we asked ourselves the following questions. Whenever we evoke a memory, does it return to the state of lability and can it be modified? Whenever I remember what things I did when I was younger, where I went to spend the weekends, who I spent the weekends with, the memory can be labilized? As for what is episodic memory, many researchers say that every time that a memory is evoked, the memory is being modified. Others say that it is not known if all memory is being modified, that few details are added but the chore of the memory remains the same. Others say that when a memory is evoked it doesn't always return to the state of lability.

So what is the role of reconsolidation? When an incomplete reminder (a key) is presented to a human or animal, it triggers the memory labilization. The brain detects an error in the prediction, it detects that the information as it is stored is not useful so it has to be updated [12] [46]. This is a unique opportunity to modify a memory that has already been stored. For example, to improve it. Reconsolidation can also be used to erase a memory. A really interesting use case is in the treatment for phobias. Also, another use of reconsolidation is in education: when a memory is labilized, adding new content occurs in a faster manner than if it is learnt from scratch.

In this work we will be studying the aspect of reconsolidation which may be involved in the formation of false memories and may even modify the declaratory part of a person in a trial.

2.2 False memories

False memories are memories of events that never occured or memories of events that are remembered differently from what actually happened [27]. The memory can suffer a partial distortion, or even an entire memory of a specific event which never happened can be implanted [29] [16] [48].

There are several theories that try to explain how false memories could be formed. One is the activation and monitoring theory [33]. This theory has two basic principles: memory activation principle and source monitoring principle.

1. The first one occurs when, for example, a person is given a series of related words (bed, pillow, tired, dreams) which are also related to a non-presented word such as sleep. Experiments have shown that people tend to think that sleep was a word which was presented in the first list, thus making it clear that a false memory was formed. So there is a tendency to remember or recognize false information as if it were true [43].

2. The other principle, the source monitoring principle, is the process that determines the source of the information of interest. Where did I get the information? If I can not distinguish well, I could form a false memory.

After some criminal offences, id lineups (or recognition wheels) are carried out. An id lineup is an investigation procedure with the purpose to identify the author of a criminal act, either by the victim of the crime or by a direct witness. The practice is only necessary when there is no previous relationship between the victim or witness and the perpetrator of the crime, so he or she cannot provide specific data (nickname, relationship, professional whereabouts, etc.) that could be used for identification. The problem after criminal offences is that usually the photos of all suspects appear on TV. Witnesses of the crime see the culprit's face in the crime scene, then weeks go by and they see suspects faces on the TV, then they put them in an id lineup and in their brain there are two types of information: the original information they saw because they were in the crime scene and the information they saw on television. Now, if a witness has an error in source monitoring, it may be that he or she believes an innocent person sawn in TV was at the crime scene when this didn't actually occur. Even a policeman who asks questions to a witness, with the questions he asks, can guide the memory to one part of the event and make the witness forget another part [8]. In fact, "corroboration of an event by another person can be a powerful technique for installing a false memory" [28]. Another false memory formation example occurs in interviews of crimes against humanity where meetings are made to get information about something. Everything that a witness says will influence the memory of another witness. This is why separate interviews should be done. In Argentina there are still no protocols for these kinds of interviews.

The more time goes by, the more contaminated the memory will become.The more time goes by, the more likely we are to generate false memories. In Argentina there are cases in which the first recognition test takes place a year later of the crime. This is something that should change after all of the scientific evidence present about false memories. There are cases of people with closed sentences with a simple eyewitness photographic recognition. It is known that 75% of innocent people with closed sentences are due to false memories (Nationwide) [37].

3 *El Laboratorio de Sueño y Memoria* and *Innocence Project*

El laboratorio de sueño y memoria is a laboratory run by Dra. Cecilia Forcato now hosted by the Department of Bioengineering of the Instituto Tecnológico de Buenos Aires (ITBA). This research environment studies how to improve memory during sleep, the transfer of information between different brain areas, integration of new information in pre-existing amnesic networks, formation of false memories, lucid dreaming, and out-of-body experiences initiated from sleep paralysis [41] .

So one of the topics the laboratory studies is what neurophysiological mechanisms are involved in the formation of false memories. We are never going to get rid of false memories because our brain works this way: forming false memories, and modifying old memories. Nevertheless, we can generate strategies to reduce their formation. So that's why the laboratory is interested in knowing the mechanism of the formation to be able to modify and reduce them.

El laboratorio de sueño y memoria paired up with *Innocence Project Argentina* in order to collaborate with a false memories formation project during sleep and wakefulness, and how these false memories can lead to wrongful identifications.

Innocence Project Argentina is one of the 67 organizations around the world which are associated in order to provide legal and investigative services for people who were wrongly convicted. This association of organizations is called *Innocence Network*. There are many causes for which a person can be wrongly convicted such as false confessions, poor defense, bad forensic science, among others, but the one that we are focusing on is the incorrect incrimination by eyewitnesses.

"As of January 2020, the Innocence Project has documented over 365 DNA exonerations in the United States. Twenty-one of these exonerees had previously been sentenced to death. The vast majority (97%) of these people were wrongfully convicted of committing sexual assault and/or murder. Although these individuals were innocent of these crimes, approximately 25% had confessed and 11% had pleaded guilty. These exonerees spent an average of 14 years in prison–10% of whom spent 25 years or more in prison for crimes they didn't commit" [38].

4 Deep Learning and GANs

Before diving into the use of NVIDIA's StyleGAN to generate faces and experiment on their features to collaborate with the laboratory, it is crucial to present the current situation and precedents regarding the technologies used for face generation. This section will therefore be about the state of the art of these technologies.

4.1 Artificial Intelligence and Deep Learning

> "The true challenge to artificial intelligence proved to be solving the tasks that are easy for people to perform but hard for people to describe formally — problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images". - Ian Goodfellow, Yoshua Bengio and Aaron Courville [14]

The field of Artificial Intelligence (AI) first solved problems that are difficult for humans to solve, problems that can be described using a set of mathematical rules, but are easy or straightforward for computers. As said by Ian Goodfellow, the real challenge was solving the problems that can't be solved with rules or operations, but feel automatic and natural for humans to solve, like understanding speech, or facial recognition. As humans we solve them intuitively, and it is very difficult to describe these solutions as mathematical rules, or explain them in any formal way. But if you think about how we solve these problems intuitively, it is because we already possess a huge amount of knowledge about the world. With this knowledge is that we behave intuitively, but it can be very subjective and therefore it is difficult to make a computer learn this knowledge too. The key challenge in AI is making a computer learn this "informal" and "non-mathematical" knowledge.

In this project we will talk about a particular solution, generating facial images and facial recognition, but take in mind that this solution can be applied to any kind of intuitive problems. When a computer learns from experience, as we do, it avoids having to specify formally all the knowledge it needs. It eventually understands the world as we do, in terms of a hierarchy of concepts, learning complicated concepts by building on simpler ones. The more intuitive the solution feels for us humans, the greater knowledge a computer needs, and greater concepts need to be built as well, with many layers between the bigger or complicated concepts to the smaller or simpler ones, and that is why this approach is called Deep Learning [44].

4.2 Machine Learning

How do computers acquire knowledge? One way is called Machine Learning. Given data taken from real-world experience, machine learning algorithms learn by extracting data patterns. They greatly depend on the data they are given, and they are designed to extract the best set of features and map this new representation to the provided data.

The most popular example of a representation learning algorithm is the Autoencoder [26]. It combines both an encoder and a decoder. The encoder converts the input data into a different representation, often compressing the information, and the decoder converts the new representation back to the original format. They are designed to preserve as much information as possible, but the main purpose is to make the new representation easier to experiment with than the original data. This new representation is obtained by separating the diversity factors that explain the observed data, or the principal components. As the model learns, it learns a new feature (edges, angles, etc.) in each layer and associates a combination of these features to a specific output. There is a layer, where the dimensionality of the data was greatly decreased and only the most important factors are left. This layer is known as the latent space, and we can imagine it as a space where different points are located, and the ones close to each other have similar values in those factors. If we think of faces again, what makes two faces similar, or two points in the latent space close to each other? A face has distinguishable characteristics (ie, skin color, eye shape, space between eyebrows). All of these can be learned by learning patterns in edges, angles, etc. Thus, as dimensionality is reduced, "foreign" information that is distinct from each image (i.e. image background) is "removed" from our representation of latent space, since only the most important characteristics of each image are stored in latent space representations.

By using representational algorithms we can study with more detail how the faces are represented, and which attributes are the ones that the algorithm uses as a representation, and which ones do not. These factors will then affect someone's perception differently, on a different scale. This project's main goal is to provide ways to study all the variables and how they can influence someone's facial recognition, specifically to see how the recognition can be tricked, and therefore, support the formation of false memories.

4.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were introduced in 2014 by Ian Goodfellow and his colleagues [17]. Their goal is to synthesize artificial samples, such as images, that are indistinguishable from authentic samples. Nowadays, there are a lot of GANs implementations, and in the image generation field, the most popular ones are the ones that excel at the formation of high quality resolution images. In this project we will use StyleGAN, trained with facial images, but before explaining how the StyleGAN network works, let's see how a GAN works.

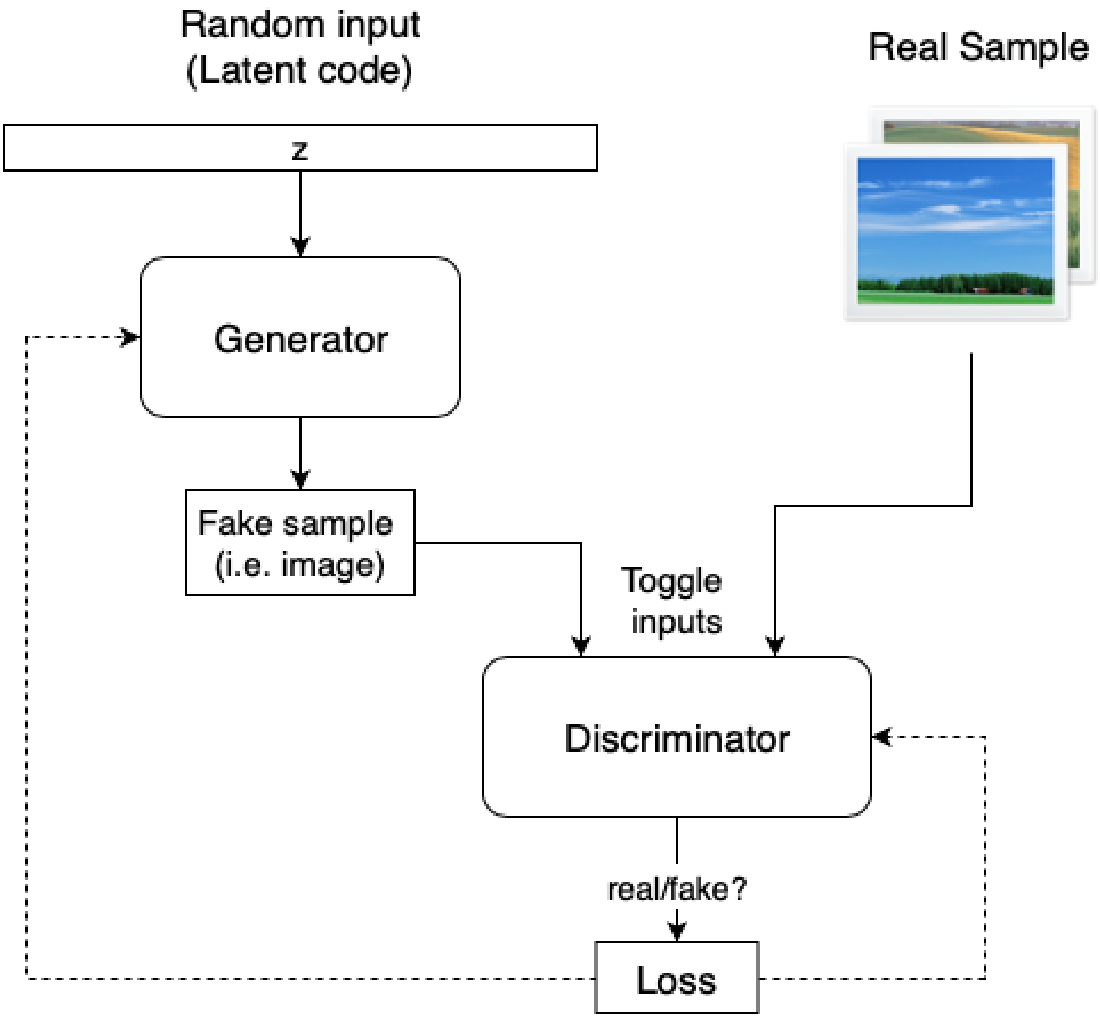


Figure 4.1: GAN architecture

A GAN network consists of the interaction between two networks. A Generator network and a Discriminator network. As its name implies, the Generator network is in charge of taking noise as input, for example a vector of a simple distribution (such as Gaussian or uniform) and transforming that noise into a sample of the distribution that we are looking for. For facial images, the noise would be transformed into an image of a face because we are looking for the "true" distribution of faces. The Generator network would be the transformation from one distribution to another, where the other distribution is very complex. But how does the Generator network learn? The other network serves precisely that purpose. The second network, the Discriminator, will receive inputs that are images and its main task will be to discriminate what is real and what is false. The discriminator will start out being quite bad at its job but once trained it will understand what is real and what is false (or try to understand it, because the generator will try its best to confuse it). The output of the Discriminator network of classifying the output generated by the Generative network will serve as feedback to learn how to improve its generations. The output of the discriminator represents the probability that the input is true, so it will be a number between 0 and 1. 0 would indicate that it is false, 1 real and 0.5 would imply that the discriminator does not know how to distinguish. The generating network will want to generate faces as realistic as possible in such a way that the discriminator's confidence about what is real and what is not decreases. Ideally, the discriminator should return "I don't know" (or 0.5) both when a user asks for real data or when he asks for generated data.

4.3.1 ProGANs

Although GANs seem to be a very fine design, the first implementations had the same problem: it took too long to train both networks in order to obtain realistic results. One of the first and many implementations of GANs, ProGANs [22], introduced a key innovation to make this easier: Progressive Training - training the generator with a very low-resolution image (4×4) and adding a higher resolution layer every time (being 1024x1024 the last layer). Initially learning a simple problem before progressing to learning more complex problems. This meant lower training time for lower resolutions, and using the lower-resolutions training in higher-resolutions training too. Overall, training time decreased significantly, and high resolutions results were achieved.

Although progressive training meant deconstructing the GAN architecture and train multiple times by layers, this architecture was only used to speed up training. Authors of a Style-Based Generator Architecture for Generative Adversarial Networks (StyleGAN) saw the innovation in ProGAN and further used it to explore the latent space in each layer. As said by the authors:

> "Yet the generators continue to operate as black boxes, and despite recent efforts, the understanding of various aspects of the image synthesis process, e.g., the origin of stochastic features, is still lacking. The properties of the latent space are also poorly understood, and the commonly demonstrated latent space interpolations provide no quantitative way to compare different generators against each other." [24]

In the next section, the mentioned aspects of the generation model will be addressed, along with how StyleGAN was built to improve them.

5 Generation Model

GANs have many and diverse implementations, but we needed one that showed state of the art results in the generation of high quality resolution facial images, because the most important goal for the project is to generate faces that look authentic and be of use in the lab studies. Furthermore, the generation model needed to be one that allowed users to navigate through the latent space, rather than maintain it “hidden” or “latent”. This section will explain how StyleGAN met these requirements.

5.1 A Style-Based Generator Architecture for Generative Adversarial Networks

A Style-Based Generator Architecture for Generative Adversarial Networks (StyleGAN) was created in 2018 to address the problem mentioned: use latent space interpolations to allow the experimentation of coarse features (pose, face shape) to fine details (hair color) to generate high resolution artificial faces. NVIDIA researchers made this happen by proposing great changes to the original ProGAN architecture, and making use of the progressive training. StyleGAN exploits the potential features of the ProGAN generator multilayer network architecture to allow control of visual features: the higher the layer (and the higher the resolution), the more detail is affected by a change, and the more important features it affects. At lower layers, for lower resolutions, greater is the change in a feature, and coarser it looks in the image. Authors categorize the effects on features according to resolutions in the following way:

1. Coarse resolutions (4x4 – 8x8): high-level aspects such as pose, general hair style, face shape, and eyeglasses.

2. Middle resolutions (16x16 – 32x32): smaller scale facial features such as hair style, eyes open/closed.

3. Fine resolutions (64x64 – 1024x1024): color scheme and microstructures.

The innovations introduced by StyleGAN have a lot to do with the generator network, and the picture below illustrates how it changed in comparison to a traditional GAN generator network. We will discuss in more detail the changes involved in the mapping network and synthesis network, and the noise introduction at each resolution level of the synthesis network.

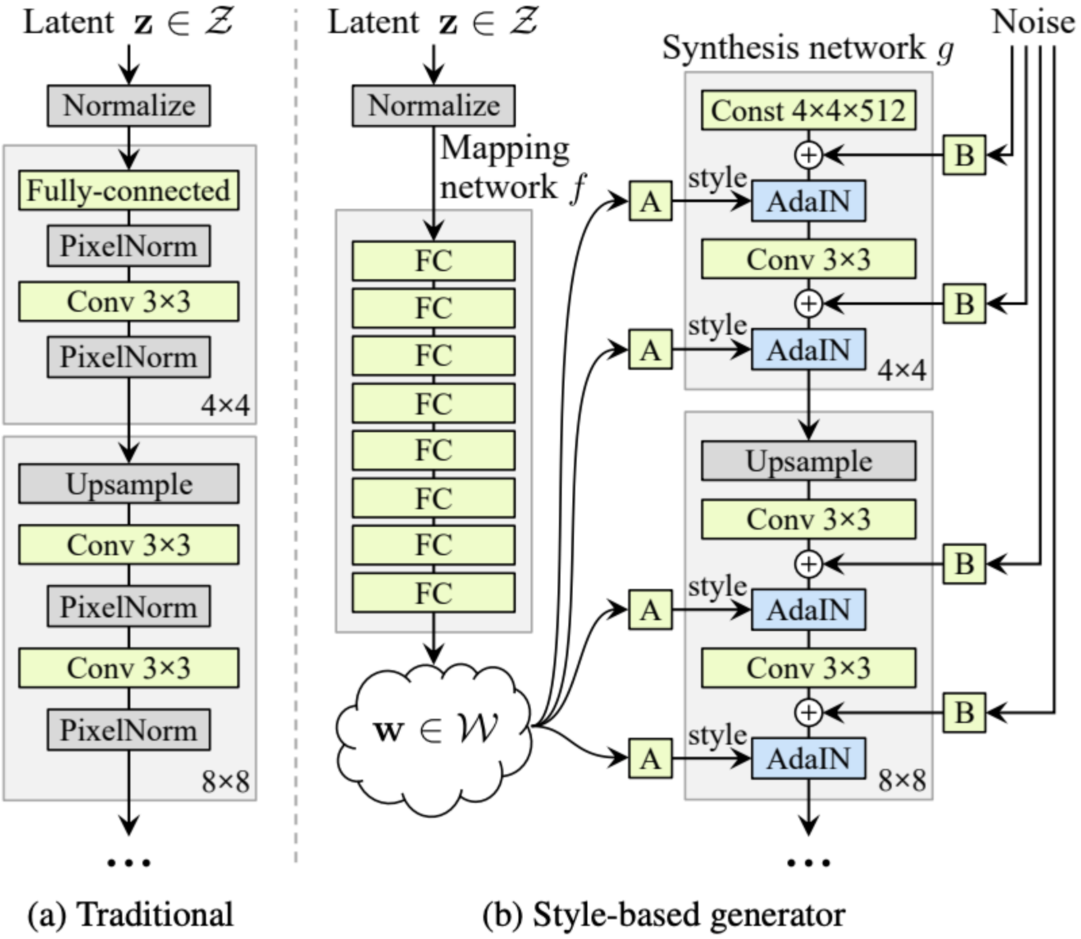


Figure 5.1: StyleGAN architecture. While the traditional generator only feeds a latent code as input, the StyleGAN generator network uses two networks: a Mapping network to map the input latent code (z) to an intermediate one (w), to later control the style at each layer (from 4x4, up sampling to 1024x2024) of a Synthesis network, to obtain an RGB image with a 1024x1024 resolution as the output of the last layer. Every component of the StyleGAN generator is detailed at [24].

5.1.1 Mapping network

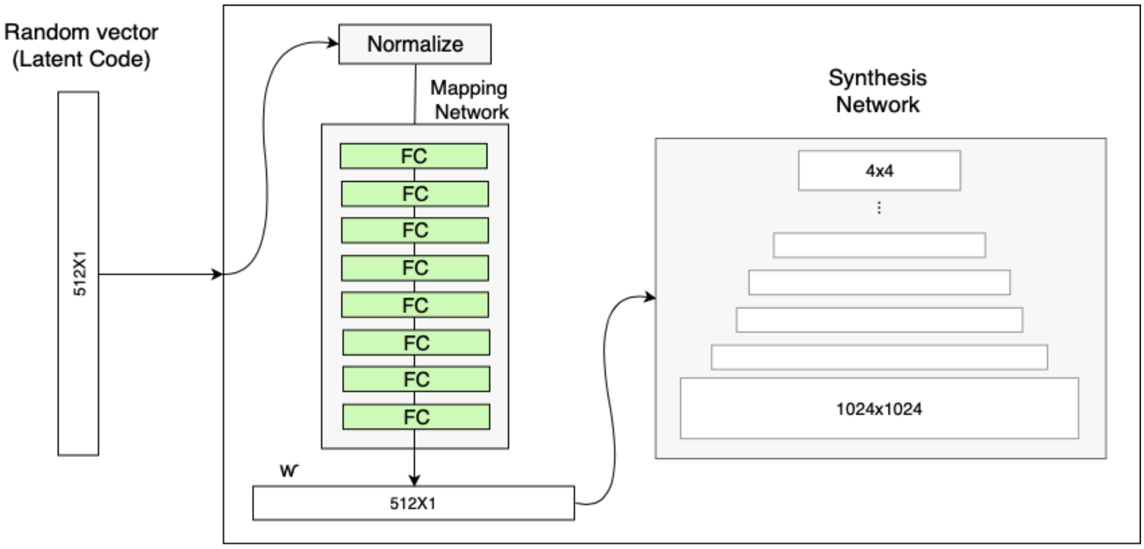


Figure 5.2: Mapping Network: mapping a vector of 512x1 dimension to another vector (w', or w in figure 5.1) in an intermediate latent space (W', or W in figure 5.1), of the same size, to later control the synthesis network. This mapping will allow the control of visual features.

[15]

The ability to control visual features is very difficult in any GAN architecture because it greatly depends on the training dataset. The traditional GAN does not allow for control over finer styling of the image because it follows its own distribution. For example, if the dataset has a major number of facial images with red hair, then there is a great probability that input values will be mapped to that feature. The only control a user has over the visual features is changing the input value (latent vector z, as seen in 5.2) and obtaining a different generated image. Therefore, if the dataset contains a majority of males with short hair and females with long hair, changing the input to obtain females with short hair would result in a change in gender, because a male with short hair is most likely to be generated. In order to obtain more control over features and styles, StyleGAN introduces another network that allows them to be independent to the training dataset's probability distribution and generate an input vector whose elements are not correlated to the dataset's features.

The mapping network maps points in the latent space (input) to another latent space (output), which have the same size, the generator uses to control style at each resolution layer. It encodes the input vector into an intermediate vector whose different elements control different visual features.

5.1.2 Synthesis network

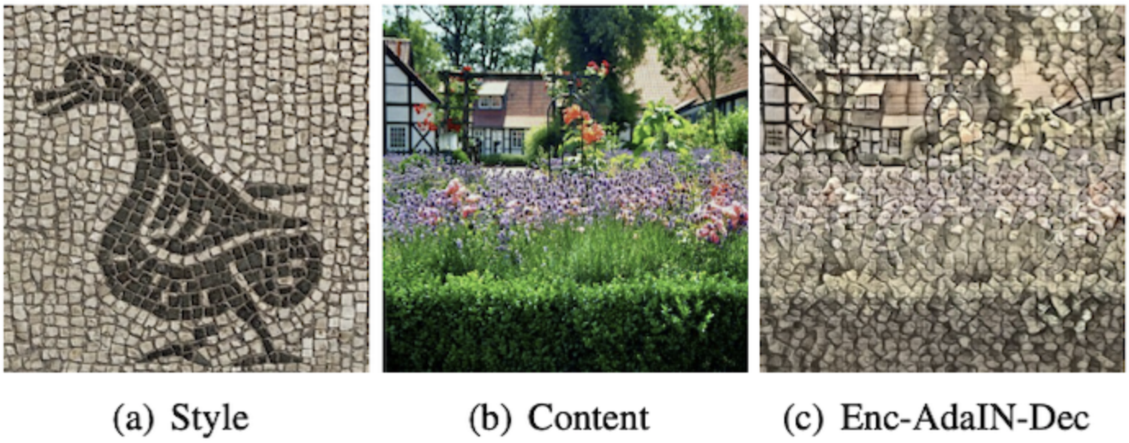


Figure 5.3: Adaptive Instance Normalization (AdaIN): receives as input an image (b) and a style (a) (the intermediate latent code, w, transformed into a style), to obtain (c) as a result of applying the style to the image.
[15]

Another key variation in the generator's network is the introduction of a synthesis network. The synthesis network works like a decoder: it converts the information obtained from the mapping network to the generated image, this being the actual output of the Generator network. In the above image the letter A is understood as a layer from the mapping network to the synthesis network. This refers to the learned affine transform: it transforms the intermediate vector W into a scale and bias for each channel of the convolutional layer. It specializes the latent code W to a style $Y = (Y_s, Y_b)$ that controls the adaptive instance normalization (AdaIN). The AdaIN module then receives a content input (b in 5.3) and a style input Y (a in 5.3) and aligns the channel-wise mean and variance of (b) to match those of (a) using the scale Y_s and bias Y_b, shifting each channel of the convolutional output. As a result a visual representation of the information (Y or style) from vector W can be obtained.

5.1.3 Noise

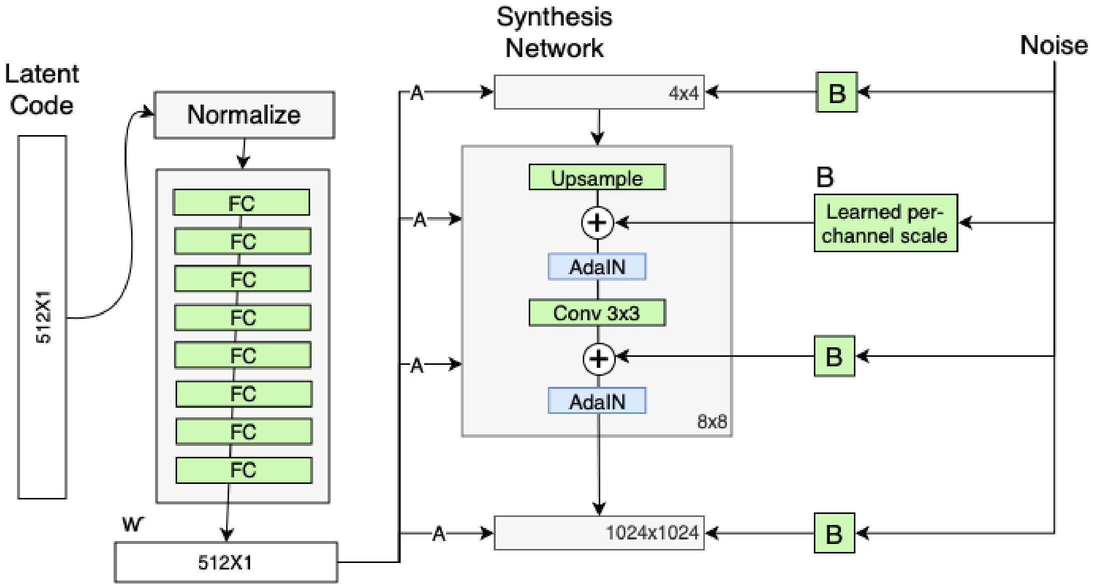
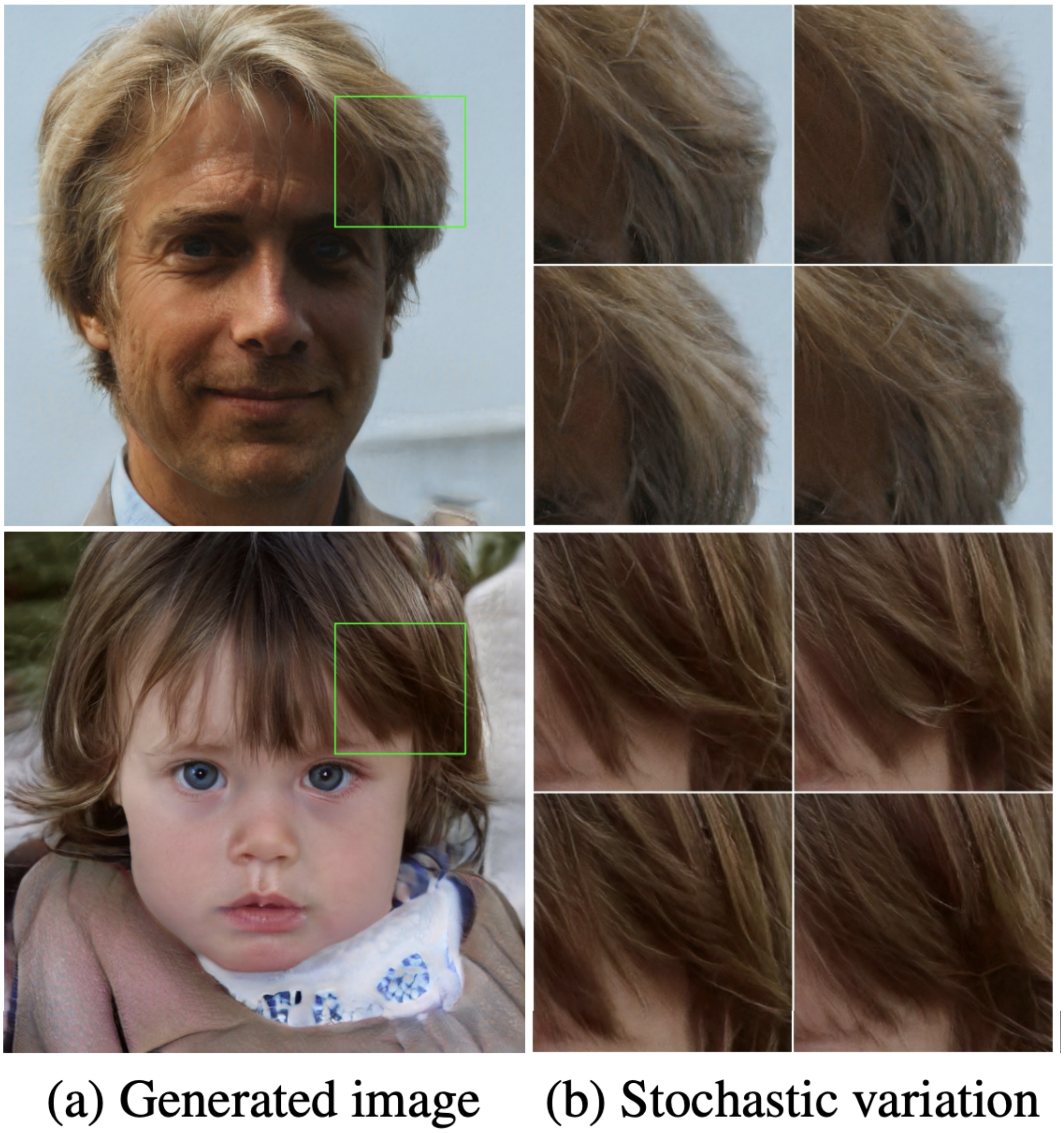


Figure 5.4: Noise introduced in the synthesis network: Gaussian noise is added after each convolution, before evaluating the AdaIN module is applied. The noise image is broadcasted to all feature maps using learned per feature scaling factors (B) and then added to the corresponding convolution. Further details can be found at [24]. [15]

To make faces look as realistic as possible, StyleGAN took into consideration stochastic variation. As seen in figure 5.4, at each resolution level of the synthesis network noise is added before the AdaIN transformation takes place. This meant to change in a small scale (but at the resolution level it takes place) the visual representation obtained from the synthesis network.



(a) Generated image

(b) Stochastic variation

Figure 5.5: An example of a stochastic variable like hair placement [24]

These noise inputs are single-channel images consisting of uncorrelated Gaussian noise, used as stochastic latent variables, that are fine visual details on a facial image such as freckles, wrinkles, specific hair direction (as seen on figure 5.5), which at a high resolution can be observed and helps to generate a more realistic output for experimentation.

5.1.4 Style mixing

StyleGAN's authors wanted to make sure that the new network did not learn the correlation between resolution levels of the synthesis network, so the generation model, during training, randomly selects two input vectors and uses the intermediate vector for the training of some levels and randomly shifts to the other vector for the remaining levels. As a consequence of making resolution levels not correlated, the generator can combine two different images and take low/middle/high level features of one image and different level features of the other image and result in a different image with both styles combined. This is what the authors call style mixing: “the ability to control the generated images, by feeding a different latent w to different layers at inference time”, and what we will most take advantage of using StyleGAN. Style mixing allows us to experiment with the generation of similar images or faces, by selecting two faces and combining different levels of features, starting from coarse styles, then middle styles and finally fine styles, the generator can show the results at each level, resulting in a transition from one face to the other. By mixing some styles, but not all at once, faces generated by the transition will look similar to the selected face but will have some different features (features that actually come from the other selected face). This will be useful for experiments if the transition can be controlled, by selecting the amount of faces to generate during the style mixing, to see how the style mixing is done gradually, and following the order mentioned before. If the amount of faces selected is very high, then the transition will be very slow, and every small change in the faces will be appreciated. If the amount of faces is too small, then the transition will not be able to show a coherent combination between the styles and the faces will not look similar. This amount should be selected according to the faces chosen to combine: if one face shows an aged woman, but the other is a young male, then the styles are too different and the transition should show a big enough number of faces to appreciate all the changes at the different resolution levels.

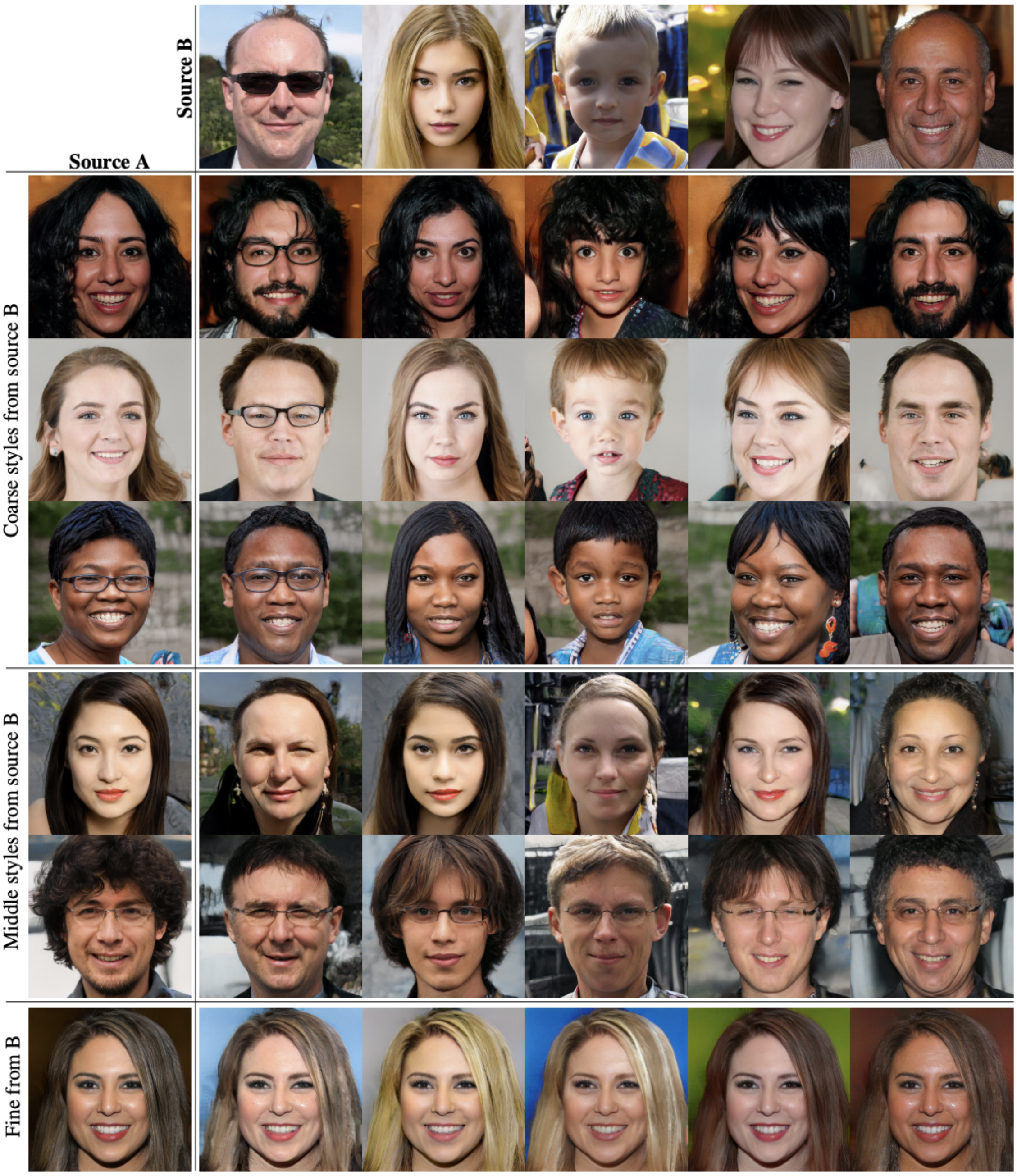


Figure 5.6: Style mixing. Two sets of images generated, sources A and B, and the combination of each image is shown. The styles combined from each source, A and B, are divided into subsets of styles: fine, middle and coarse styles, depending on the resolution of the styles. The combinations are a result of copying those subsets of styles from B, and copying the rest from A.

[24]

As a result, multiple images can be combined in a coherent way. Combining two images I_1 and I_2, the result takes some features from I_1 and the rest of the features from I_2. For example, figure 5.6 presents examples of images synthesized by mixing two latent codes at various scales. It can be seen that each subset of styles controls meaningful high-level attributes of the image.

5.2 Analyzing and Improving the Image Quality of StyleGAN

StyleGAN2 offers great improvements to the original architecture. The main changes made, like reconstructing the Adaptive Instance Normalization as Weight Demodulation and the removal of progressive training (the model is not explicitly required to have different resolution levels) showed great results at faster training and higher quality results. The authors noticed in the first version of StyleGAN that the network had a strong location preference for fine features like nose, mouth and eyes, and that was attributed to the progressive training, as the network first trains with low resolution images and then scales it up when a convergence property is met. The removal of progressive training and introduction of a residual-nets skip connection [21] between lower resolution feature maps to the final generated image showed that the higher resolution level (1024x1024) contributes more in the results, giving more contribution to fine features than earlier. This means that they can also be more easy to change too, using style mixing.

5.2.1 Path length regularization

One of the most interesting additions in the last version is the path length regularizer, with the intention to improve the quality of results, but with a very beneficial side effect to make the generator easier to invert. The path length regularization is the preservation of the vector's length, regardless of the direction. Given a latent vector w, a fixed-size step in w results in a non-zero, fixed-magnitude change in the image. As said by the authors,

> "We can measure the deviation from this ideal empirically by stepping into random directions in the image space and observing the corresponding w gradients. These gradients should have close to an equal length regardless of w or the image-space direction, indicating that the mapping from the latent space to image space is well-conditioned." [25]

During experimentations, the path length regularization showed that the latent space and the whole architecture is easier to explore. By making arithmetical operations on a latent vector and generating the output image we could explore the latent space and the resolution levels by making small directional changes to the vector. This was the key to achieve the previously mentioned "transition" from one face (one vector) to another. Given the amount of images to generate ($amount$), a starting position ($startingVector$) in the latent space, and a destination position ($destinationVector$), we could account for the new vectors in the transition by finding the x in a very simple linear equation:

$$destinationVector = amount * x + startingVector$$

Solving this meant that we could use a fixed-step in w and generate the fixed-magnitude change in the face.

Further changes were made to the architecture, but all previously mentioned features in StyleGAN were the ones we most explored and used for the implementation.

6 Face Generator

This project aims to build an interface that allows adequate navigation of the latent space generated by StyleGAN and generate a personalized output controlled by the user. This will require complying with all the computational, software and hardware requirements, which StyleGAN implies, and at the same time, making it available to the laboratory.

6.1 Solution Design

Our solution consists of developing a web application that is hosted in the Titan GPU provided by ITBA, specifically the NVIDIA Corporation GP102 [TITAN Xp], where the StyleGAN environment is set and the training of the network and generation of results can be obtained via HTTP requests. This web application will be an API Rest that will serve as an interface to the methods that can be used from StyleGAN2, using it as a library, with a fixed pre-trained network. For the laboratory, another application will be developed in order to have a layer of usability more appealing to the researchers at the lab. These two applications will work together, the front-end application consuming the StyleGAN API, or the back-end application, and ultimately the user will only see and use the front application at the lab. The back-end application will be consulted via HTTP requests to perform the different functionalities, and provide the necessary responses. Besides using the StyleGAN network, the back-end application will persist the faces generated by the network in a in-memory database, to keep track of all the faces the user will be able to experiment with and eventually ask for similar faces of a particular previously generated face.

The user will generate images from the pre trained network, and use the transition from one face (let's call it the "original" one) to another (the "destination") to make use of the style mixing properties of the latent space and generate similar faces of the original face, while also controlling which features to mix and which to maintain the same (or almost the same), by selecting the destination face to which the original face should transition to.

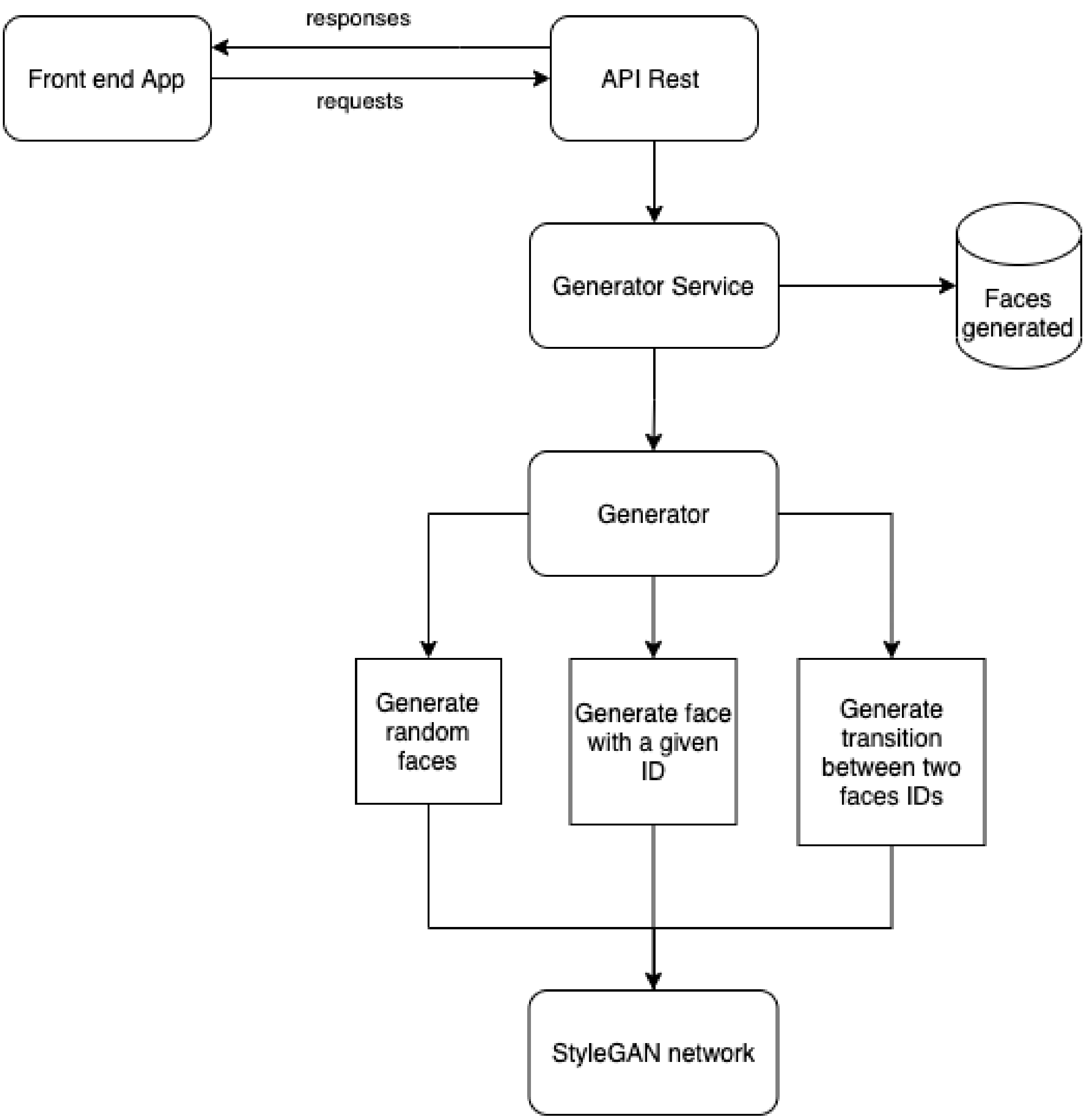


Figure 6.1: Application flow: front-end application consuming through HTTP the API Rest for the generator services, including the generation of random, specific and transition faces, achieved using the StyleGAN network. The services use the database to persist the identification of the generated faces, to allow the use of previously generated faces for the services.

6.1.1 Functionalities

- Generate an amount of random facial images.
- Generate a specific facial image, identified with an ID.
- Generate a transition between the "original" face and the "destination" face to obtain similar faces of the original one.

6.1.2 Quality attributes

Before designing the architecture, the quality attributes were defined to prioritize some of the points of view and features of the solution.

1. Precision: results should be of a very high quality resolution and should not appear to be artificial. The user should not be able to distinguish the generated face from a real one.

2. Performance: the solution is expected to perform its functions within certain given restrictions, such as speed, accuracy, or memory usage.

3. Maintainability: the solution is expected to be able to be subjected to repairs and evolution.

6.2 Software

A challenge with machine learning development environments is that they rely on complex and continuously evolving open source machine learning frameworks and toolkits, and complex and continuously evolving hardware ecosystems. The frameworks and toolkits required for StyleGAN are:

- 64-bit Python 3.6 installation. Anaconda3 is recommended, with numpy 1.14.3 or newer.

- TensorFlow 1.14 or 1.15 with GPU support.

Another software requirement is the use of both Linux or Windows, but Linux is strongly recommended for performance and compatibility reasons.

6.2.1 Python and StyleGAN2's repository

From StyleGAN2's public repository at Github, the repository is downloaded and used as a library in methods and scripts programmed in Python for the API. The only script programmed to use StyleGAN2 is encapsulated in a Generator class, which needs all the input the StyleGAN2 methods need to download and initialize the network in Tensorflow. Once the Generator class and the network are running, the following methods can be used:

- *generate_random_images*: receives as input the number of images to generate, and a random seed to generate those images from.
- *generate_transition*: receives the seed corresponding to the original face, the starting point of the transition, and the seed corresponding to the destination face, to which the transition will be directed. It also receives a speed scalar (*speed*) and a number of faces (*qty*) to generate in the transition. With these two parameters, the step between each face generated in the transition is defined as: $step = (seed_{to} - seed_{from}) * speed/qty$

A service layer was programmed for the web application to consume. It instantiates the Generator and the database, to persist the generated faces using the seed the Generator used, and associating them with an identification number, that will be returned to the user to ID the faces and later make the transitions with the Generator.

6.2.2 Pre-trained network and FFHQ dataset

The proposed solution does not consider the training of a custom dataset as an input. The StyleGAN network used to generate images is a pre trained network provided by NVIDIA researchers, available to download through a Google Drive link [23]. The dataset used to train this network, Flickr-Faces-HQ (FFHQ), is a high-quality image dataset of human faces, originally created for StyleGAN. The dataset consists of 70,000 high-quality PNG images at 1024×1024 resolution and contains a high diversity in terms of age, ethnicity and image background. It also considers accessories such as eyeglasses, hats, etc. The images are from a website called Flickr, available to store and share photos, thus the dataset contains all the biases of that website [10]. Filters were used to prune the set, to remove the occasional statues, paintings, or photos of photos. Finally, the dataset was cropped and aligned and used to train the network.

6.2.3 SQLite

The database is an in-memory database using a SQL motor called SQLite3. This reduces latency in accessing the database, as it is part of the application instead of using a server-client database hosted outside the application's environment. As mentioned above, it contains a table that associates an ID with a Seed as columns, to persist as a row every face generated with the seed as a unique long number, and the ID as a primary key. When a face is identified, the seed can be consulted by connecting to the database, and later mapped in the latent space using the Generator's methods.

6.2.4 Flask and Gunicorn

Flask is a Python framework used for web applications. Using Flask, an API Rest was developed with endpoints that use the Generator Service. The following endpoints are declared:

- GET $/faces$: to obtain the ids of the generated faces stored in the database.
- POST $/faces?id = \{id\}$: to generate the face of the given id and saved in the results folder.
- POST $/faces?amount = \{amount\}$: to generate a random amount of faces and save them in the results folder.
- POST $/faces?amount = \{amount\}\&id1 = \{id1\}\&id2 = \{id2\}\&speed = \{speed\}$: to generate a transition from the face of $id1$ to the face of $id2$, using $speed$ and $amount$ as the parameters needed for the transition. $id2$ is an optional parameter, which in case it is not given, a random face will be chosen to direct the transition to.

6.2.5 Front-end application

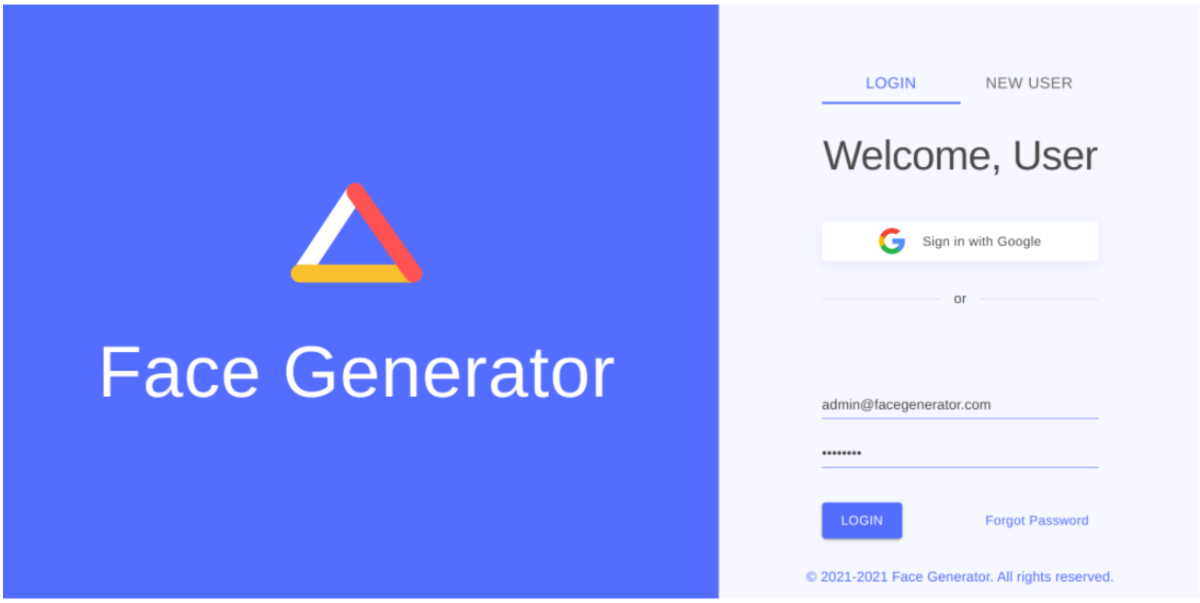
FG-Style App is the user interface to interact with the back end which generates and manipulates faces. The objective of the application is to ease the work of the scientists in *El Laboratorio de Sueño y Memoria* so they can have an intuitive and simple tool to use. The application enables users to generate artificial, realistic and high-quality facial images, and generate a transition between two artificial faces.

The primary quality attribute taken into account when designing the application is usability. Users with little experience in the use of GANs such as StyleGan should be able to use the application to generate faces and transitions.

It is also important that the application is secure and that the images generated are not filtered because this could eventually hinder the experiments done with this tool. Because the app and the server will both be running inside a computer at ITBA, additional security measures such as authentication weren't taken into account. In future iterations, the whole project could be made available and good practices regarding security should be implemented.

FG-Style App is done in ReactJS. ReactJS is an open source Javascript library, which offers great benefits in performance, modularity and promotes a very clear flow of data and events, facilitating the planning and development of complex apps. It is important to note that ReactJS is a library focused on visualization and that was something we needed. ReactJS has a very high performance, because it implements a Virtual DOM and instead of rendering the entire DOM on each single change, which is what is normally done, it makes the changes in a copy in memory and then uses an algorithm to compare the properties of the in-memory copy with those of the DOM version and thus apply changes only to the parts that vary.

On the welcome window (Figure 6.2) it will be possible to login straight away by clicking the login button because, as said before, no authentication method has been implemented. Yet, as it can be seen in Figure 6.2, future iterations on the project could implement authentication.



admin@facegenerator.com

••••••••

© 2021-2021 Face Generator. All rights reserved.

Figure 6.2: Welcome window

On Figure 6.3, the Generate Random Faces window is shown. There will be a field which accepts a numeric value to indicate the amount of new random faces to be generated. This can be a value from 0 to 30,000. The user will then have to click on the *generate* button so that the faces are generated. Today's implementation will require the user to go to the folder *face-generator/results* to see all of the faces generated with their corresponding ids. The id of an image can be extracted from the image's name. It will be the number following the word "image". So "image34.png" indicates that the id of that face is 34.

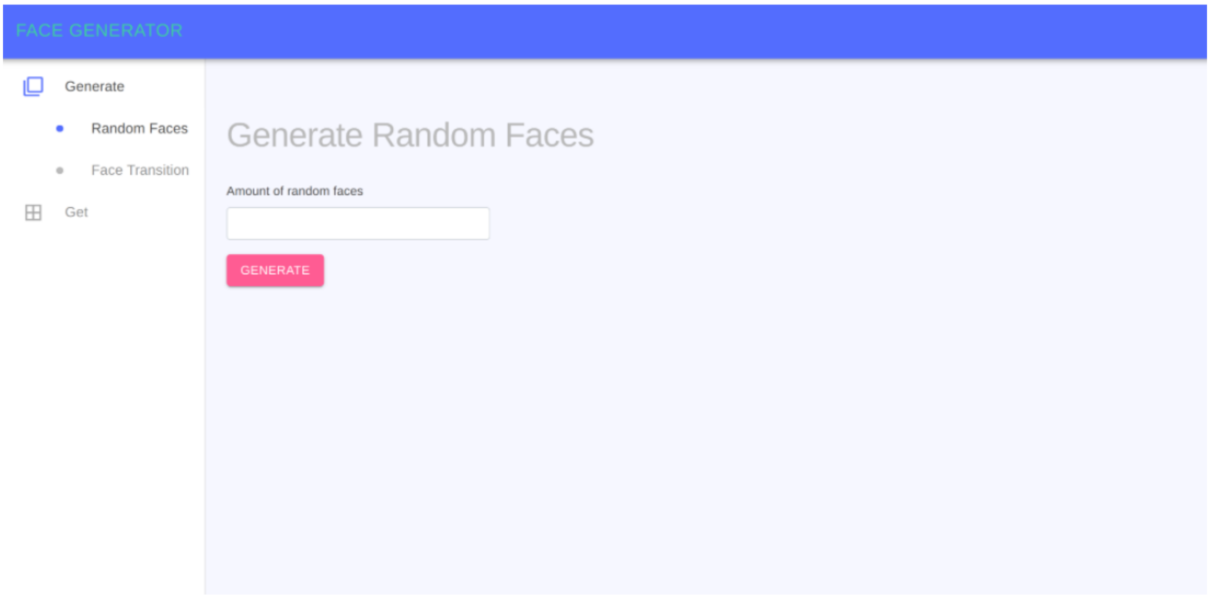


Figure 6.3: Generate Random Face window

On Figure 6.4, the Generate Face transition window is shown. This functionality requires a series of parameters:

1. **Id of first face:** This is the id from one of the images previously generated which are saved at face-generator/results. The transition will start at this face.

2. **Id of second face:** This is the id from one of the images previously generated which are saved at face-generator/results. The transition will be headed to this face from the first face.

3. **Distance from first face:** The transition will be done **from** the first face **to** the second face but will stop at a "place" between both faces. The "place" where it stops is the "distance from first face". This distance measure can be thought like a continuous variable representing degree of similarity: 0 being complete similarity with the first image, and 1 being complete similarity with the second image.

4. **Amount of faces to generate:** This is the amount of faces to generate between

the first photo and the last face determined by the previous field.

After the user inputs all of the previous fields, pressing the button *generate* will proceed to generate the transition which will be saved, in this case, at the *face-generator/results/transitions* folder. Each transition made will have a folder inside the previous folder with a timestamp unequivocally identifying each transition.

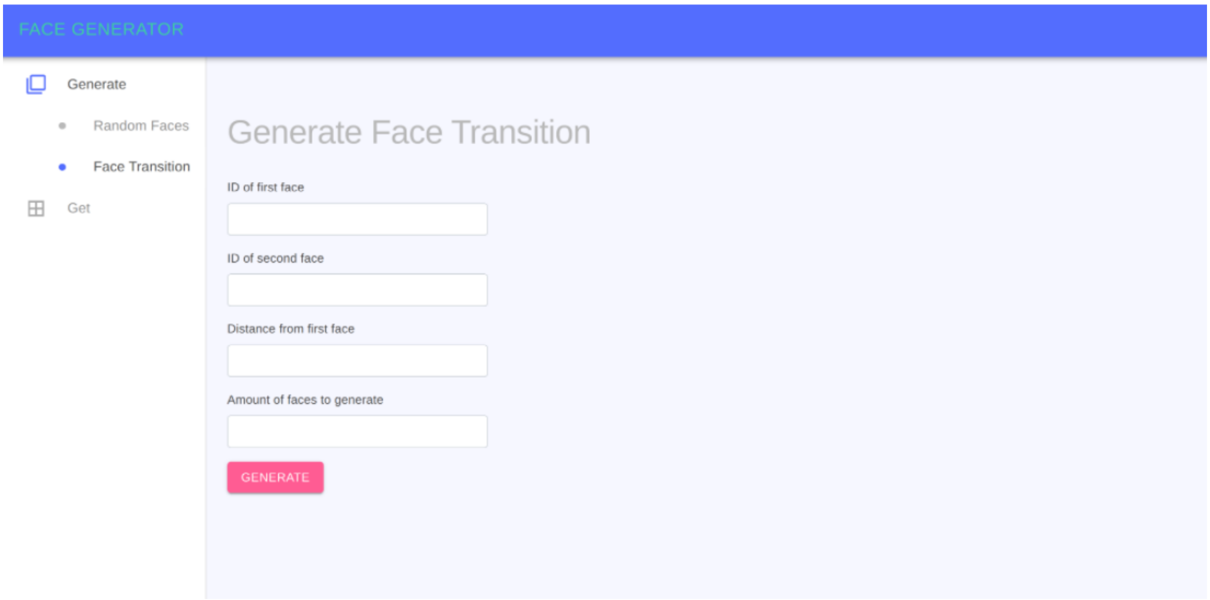


Figure 6.4: Generate Face Transition window

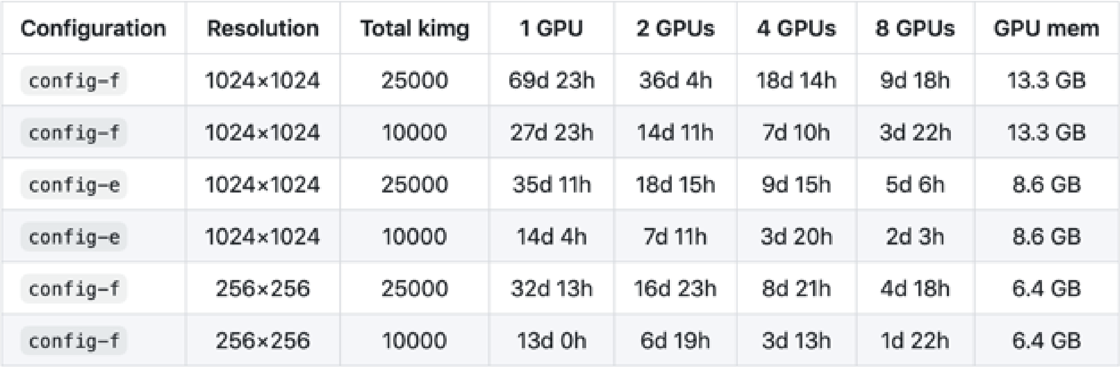
6.3 Hardware

For a machine learning development environment, one of the most important ingredients is a very powerful compute level: High-performance CPUs and GPUs to train models. Because StyleGAN is a project developed by NVIDIA, it can only work on NVIDIA GPUs. It is required that one or more high-end NVIDIA GPUs with at least 11GB of DRAM are used. StyleGAN recommends NVIDIA DGX-1 with 8 Tesla V100 GPUs. According to this research [45], comparing with other NVIDIA GPU models performance has resulted:

	Performace
V100	1x
2x P100	1.19x
P100	1.79x
GTX 1080	2.29x
K80	5.4x

Table 6.1: NVIDIA GPUs performance.

The table 6.1 shows training times for a V100 GPU. At ITBA we have an NVIDIA Titan XP GPU, that has almost the same number of CUDA cores and other specs as the GTX 1080, so training time is expected to be 2.29x as follows:



Configuration	Resolution	Total kimg	1 GPU	2 GPUs	4 GPUs	8 GPUs	GPU mem
`config-f`	1024×1024	25000	69d 23h	36d 4h	18d 14h	9d 18h	13.3 GB
`config-f`	1024×1024	10000	27d 23h	14d 11h	7d 10h	3d 22h	13.3 GB
`config-e`	1024×1024	25000	35d 11h	18d 15h	9d 15h	5d 6h	8.6 GB
`config-e`	1024×1024	10000	14d 4h	7d 11h	3d 20h	2d 3h	8.6 GB
`config-f`	256×256	25000	32d 13h	16d 23h	8d 21h	4d 18h	6.4 GB
`config-f`	256×256	10000	13d 0h	6d 19h	3d 13h	1d 22h	6.4 GB

Figure 6.5: Training time for Tesla V100 GPU.

Using an NVIDIA GPU is not the only requirement. GPU should also support:

- CUDA toolkit 10.0,
- cuDNN 7.5.0.

7 Results

This section will show some of the results achieved. Many faces were generated at random, but we manually chose to show and test the transition with these four faces because they do not contain any accessory such as glasses or hats. Image resolution has not been reduced.

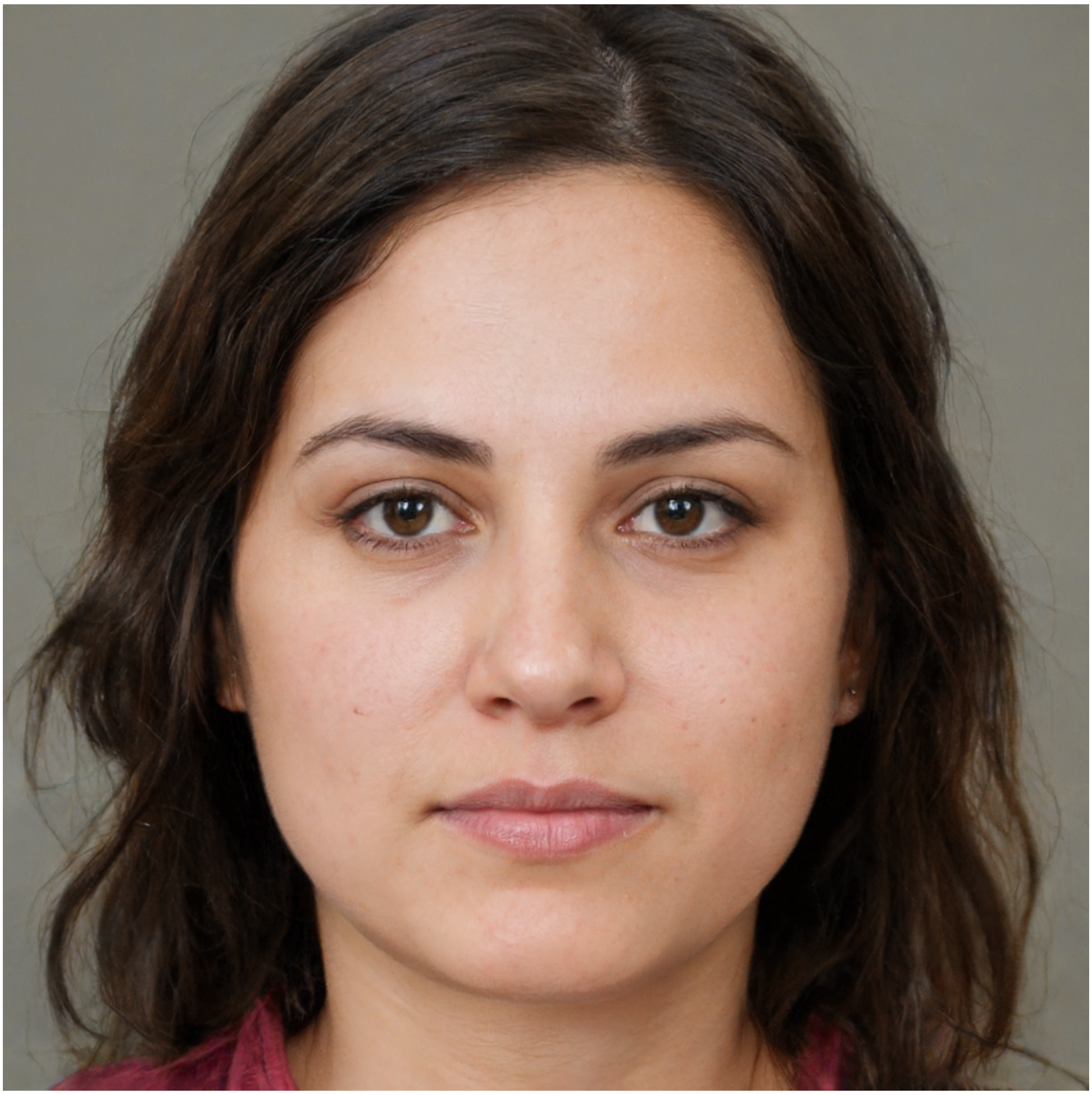


Figure 7.1: Face 1.

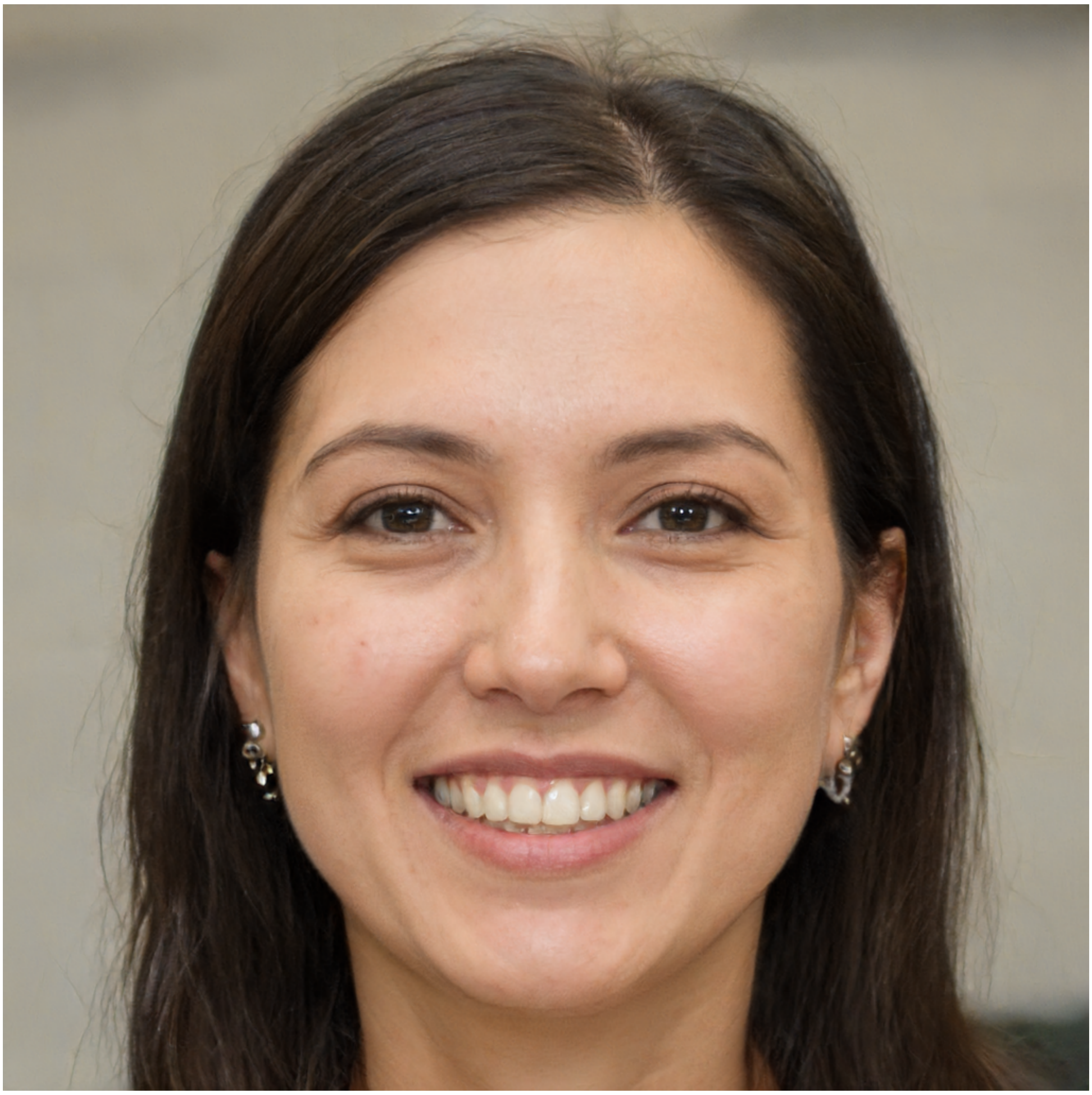


Figure 7.2: Face 2.

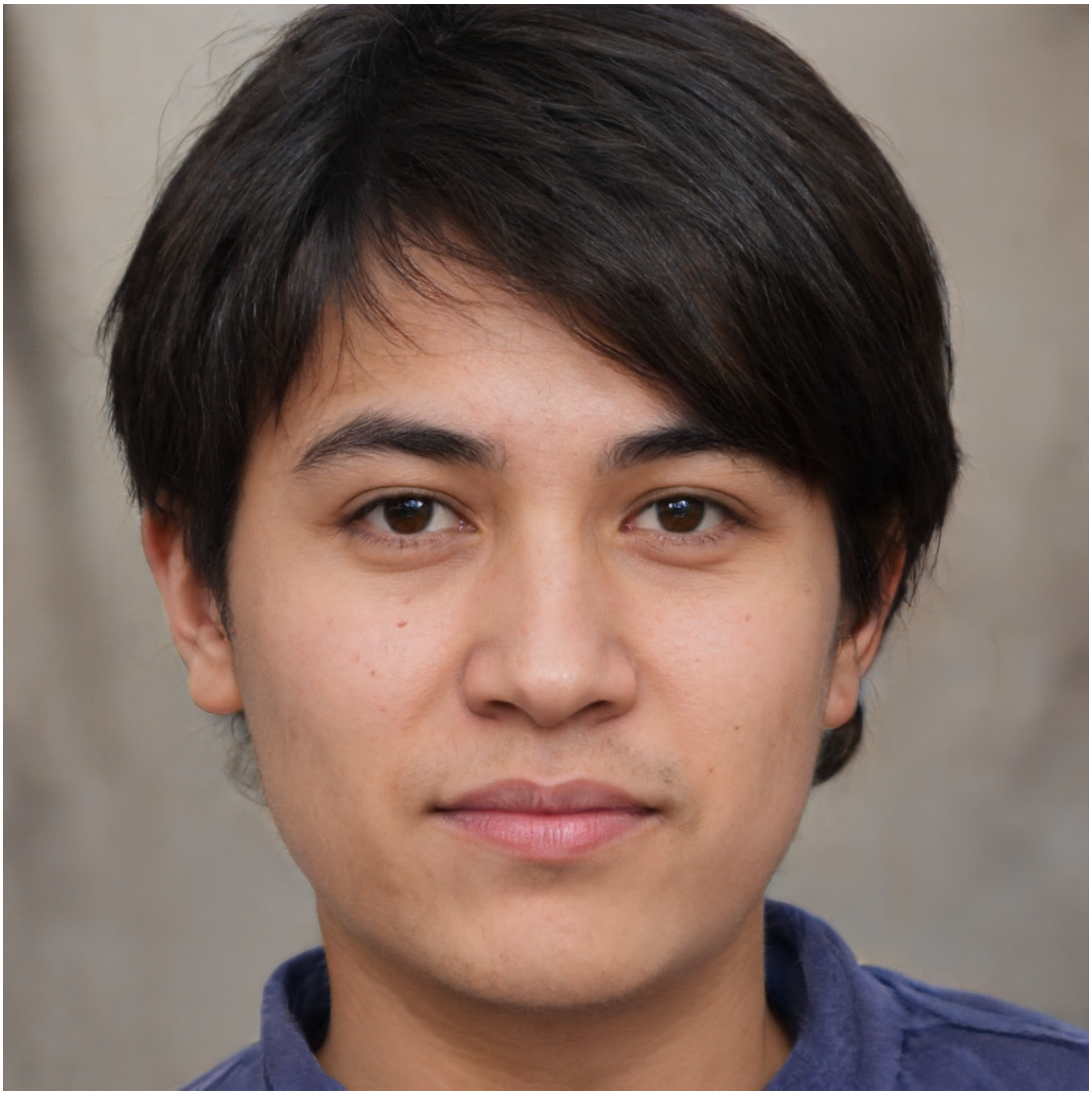


Figure 7.3: Face 3.

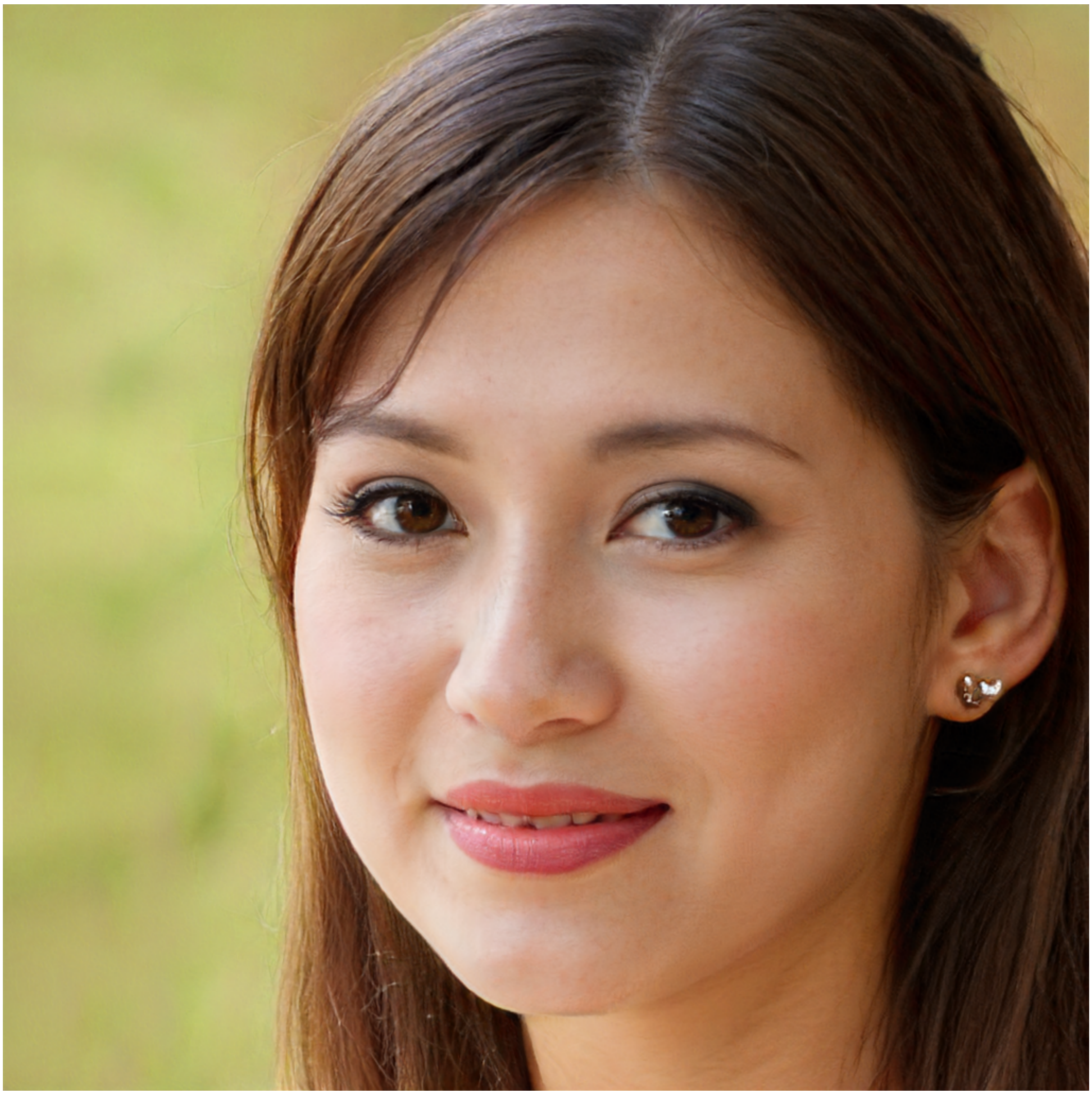


Figure 7.4: Face 4.

Face 1, in figure 7.1 was selected empirically to generate similar faces, due to the fact that it already complies with a neutral face, using the remaining three faces (in figures 7.2, 7.3, 7.4) as the destination faces to direct the transition. As these three faces are very different to each other and to face 1, every transition was done with different amount and distance parameters, in order to obtain what we thought it would produce better results. Every transition generated needs to be tested an enough number of times to achieve the best parameters, and therefore, good enough similar faces to the selected face. The goal in every transition is to generate similar faces but not equal, and that the faces generated comply with all the requirements the lab will have in the studies: neutral faces, no accessories, neutral backgrounds, looking straight, no posing, etc. Those requirements can be met by choosing the correct transition (changing the amount and distance parameters), but definitely the most difficult requirement will be maintaining the face frame. We will show these particular results to the laboratory and get their feedback.

7.1 Transition from face 1 to face 2

This transition is done from face 1 in figure 7.1 to face 2 in figure 7.2.

A video showing a very slow and full transition, with 300 faces as the amount to generate, can be seen in [30].

After watching that transition, we chose to generate a much shorter transition, with 3 images as the amount asked, and keep the distance in 1.0. We observed that the first transition had many very similar faces and that it was unnecessary to generate so many images.

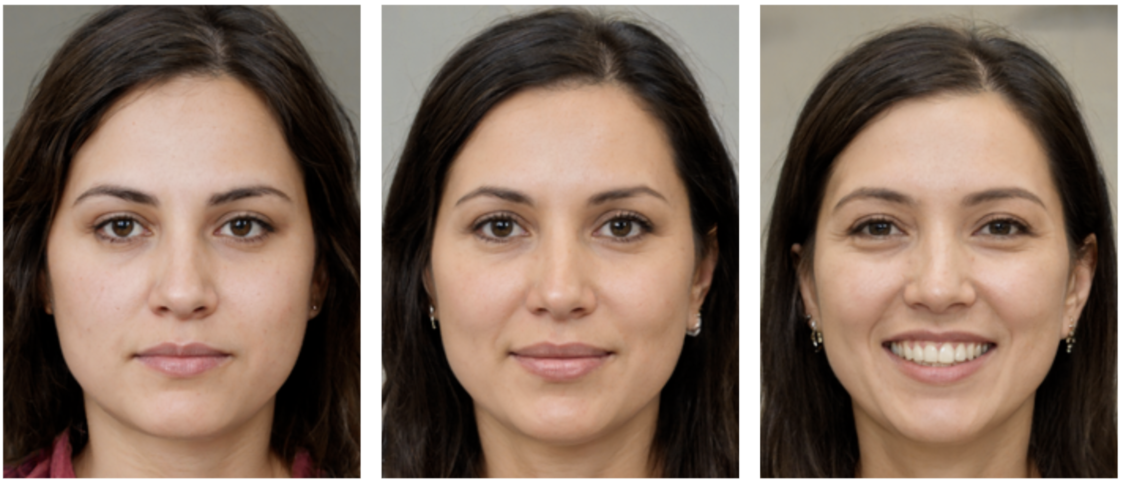


Figure 7.5: Transition 1: amount=3, distance=1.0.

For the purposes of the laboratory, we observed that making a transition to a face that is smiling does not produce the results needed, and it can already be seen that in the second face generated (the "middle" of the transition, or at distance 0.5) the face has a small smile. To get better results, we made a transition to the same face but changing the distance parameter to 0.5, so the transition will be to the second face of the transition 1.

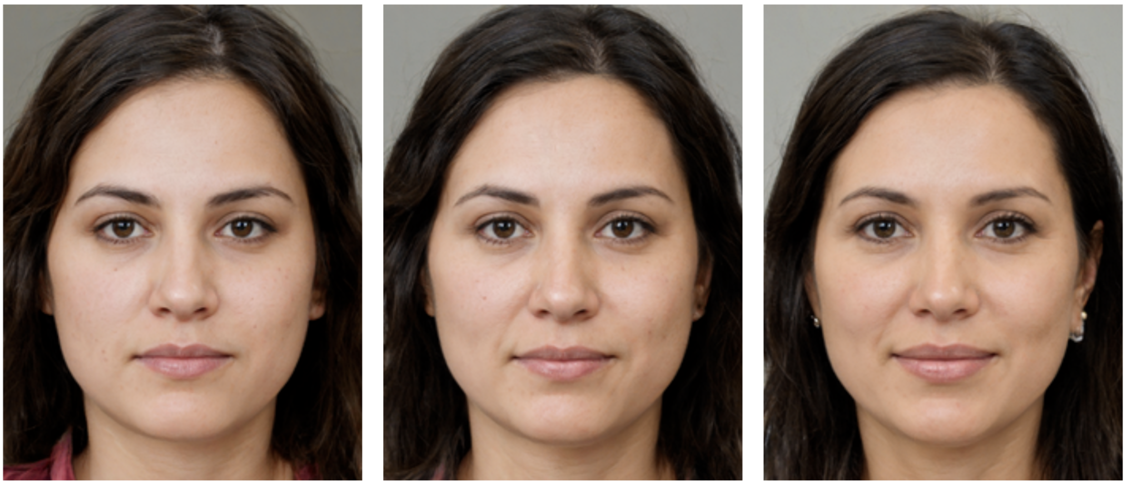


Figure 7.6: Transition 2: amount=3, distance=0.5.

If we select a shorter distance, we can appreciate better results in regards to avoiding the smile.

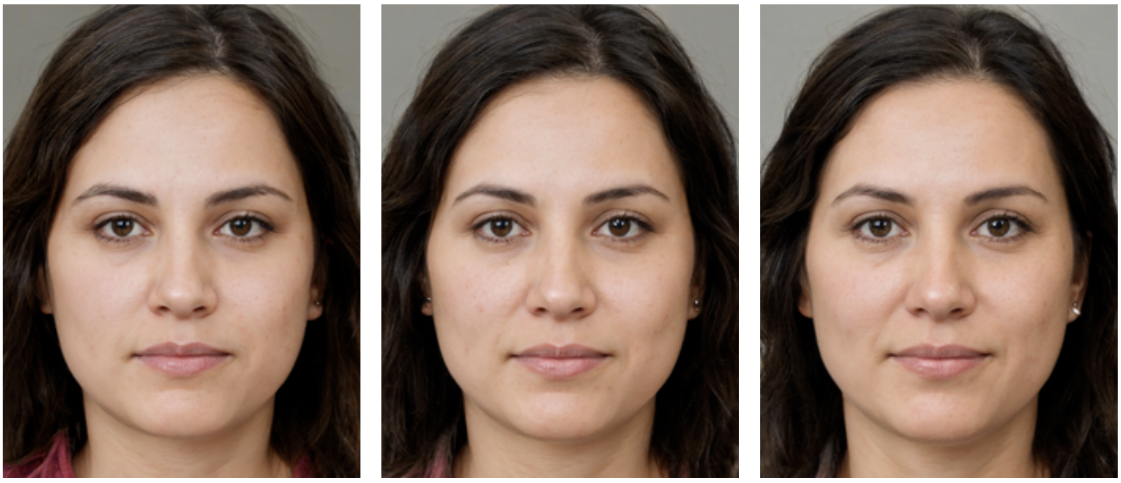


Figure 7.7: Transition 3: amount=3, distance=0.3.

7.2 Transition from face 1 to face 3

To further explore how the style mixing can be of use even when the features are very far apart, we chose to do a transition between different genders, the woman in face 1 (figure 7.1), and the man in face 3 (figure 7.3). We also wanted to observe every step in a slower transition that can be seen in [31], with amount=50, where it can be appreciated how low-level features such as hair shows a change in every small change (step) to the latent vector, while medium-level features such as skin color shows the changes made in a slower way, and high-level features such as eyes and nose show very small changes.

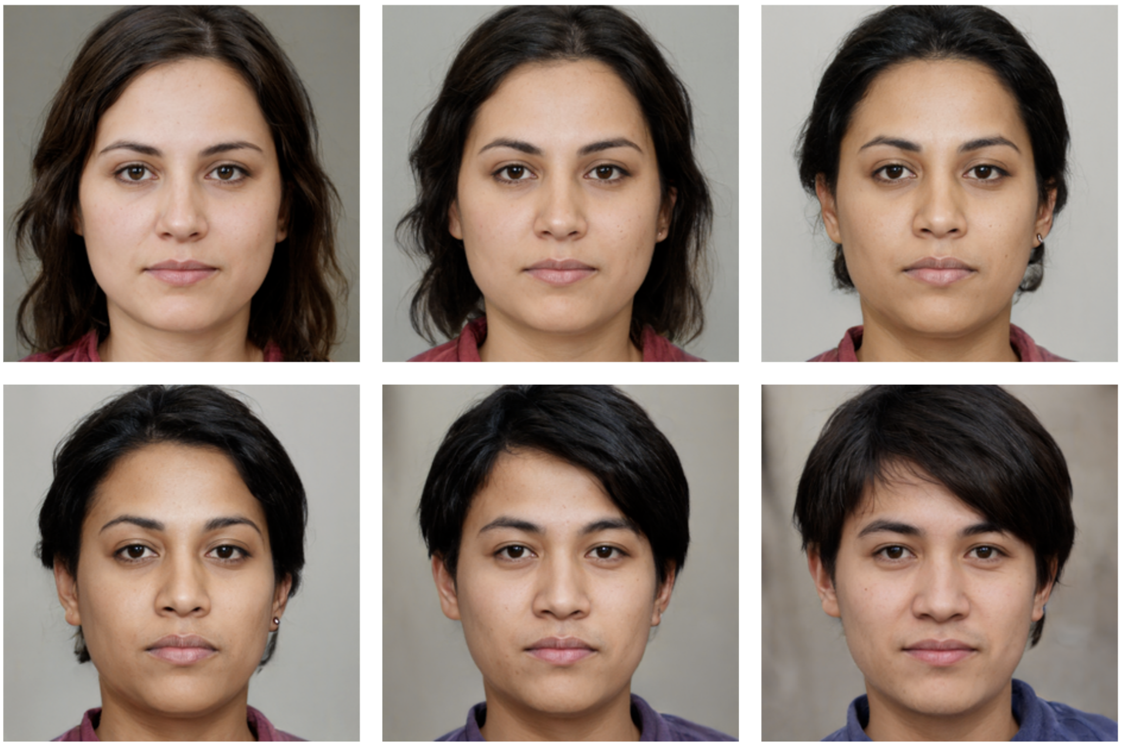


Figure 7.8: Transition 4: amount=6, distance=1.0.

7.3 Transition from face 1 to face 4

After doing the previous transition, we understood that mixing such different styles can be of use only if the distance is kept in a shorter amount. For the next transition, we decided to combine face 1 and face 4 (figure 7.4). Because face 4 is not a neutral face and is posing sideways and showing one side of the face more than the other, the transition was kept at distance 0.5 to observe which features are kept the same and which ones change.

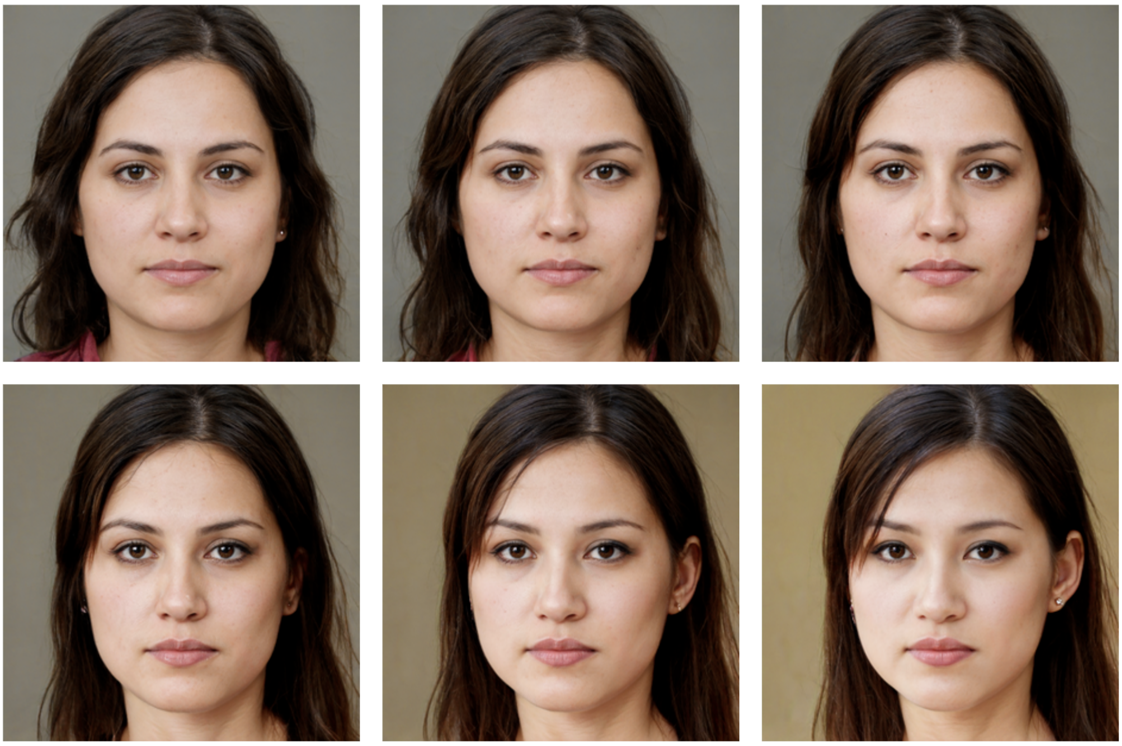


Figure 7.9: Transition 5: amount=6, distance=0.5.

A slower transition, with amount=100, can be seen in [32].

7.4 Further exploration in features

As mentioned before, the second version of StyleGAN introduced a smoother latent space using path length regularization, which enabled the inversion of the generator: projecting images into the generator instead of using a latent code to obtain the images. We explored this functionality of the generator because it would be useful for the laboratory to have as an input of the application any custom image to experiment with, rather than selecting IDs of pre-generated random faces. However, this does not avoid the restriction of having the network pre trained with the FFHQ dataset, and with any custom image the projection will be done with the training achieved using that dataset.

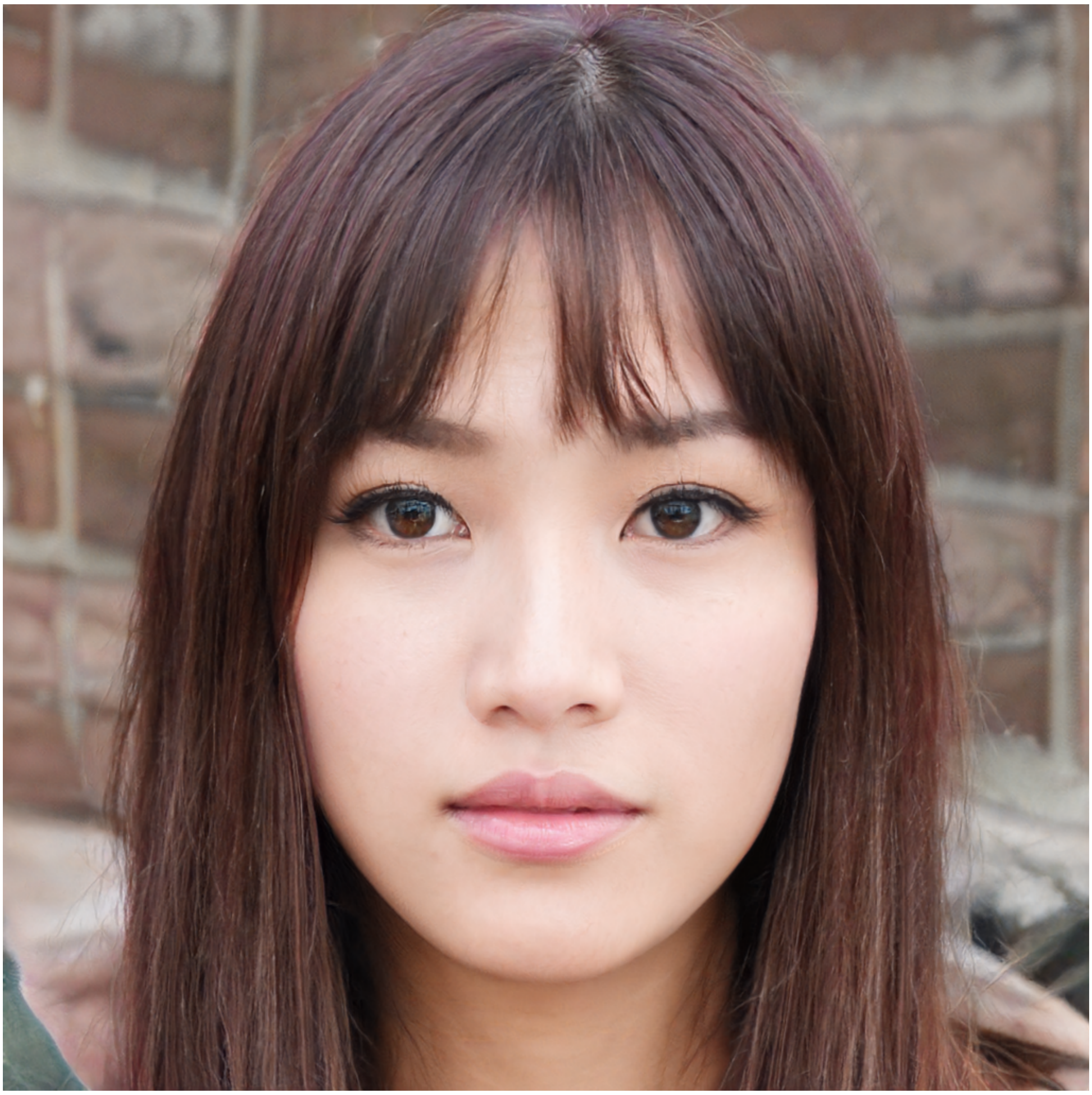


Figure 7.10: Face 5.

Figure 7.10 shows the face used to experiment with, generated at random.

The official encoder of StyleGAN2 [9] was used to obtain a projection of the generated image. We assumed that the best way to obtain the projection would be by using a generated image, with the pre-trained network, instead of a custom one that was never learned by the network.

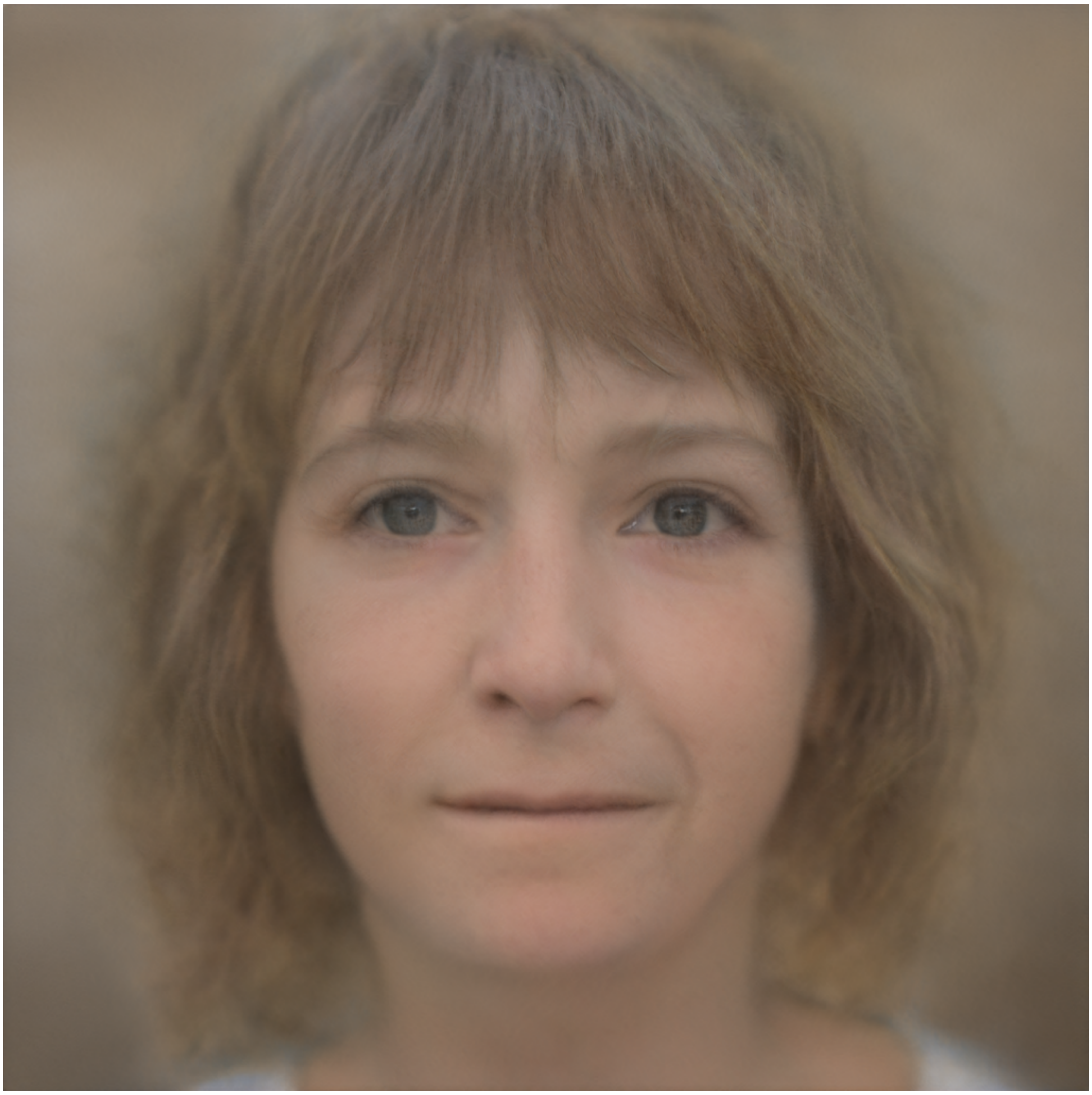


Figure 7.11: Encoded and projected face 5.

Figure 7.11 shows the face's projection in StyleGAN2. This was achieved by using the encoder with 750 iterations, the recommended amount by the encoder's authors. It can be seen that this face is not one that could be approved by the discriminator network, and should not be used in experiments. However, we wanted to further study if we could modify specific attributes of the encoded face. We obtained features from the same source as the encoder, which are latent directions used in arithmetic operations with the encoded images. By changing the obtained latent code of the face generated to the direction in the latent vector of the selected feature, and using a scalar to generate an intensity of the modification that could result, we could generate the encoded image's face with the features changed.

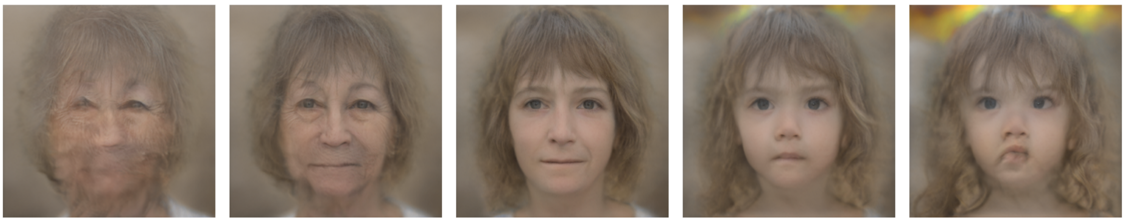


Figure 7.12: Different levels of intensity applying changes to the age feature.

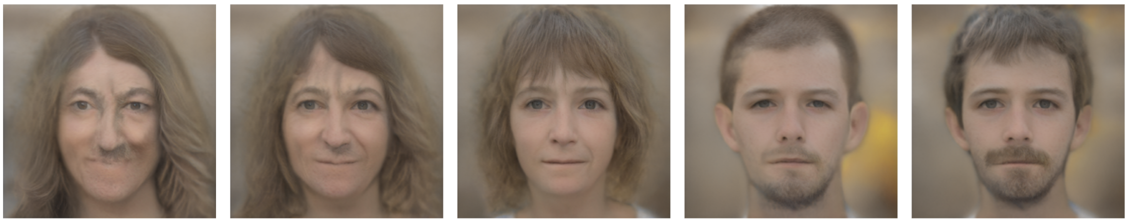


Figure 7.13: Different levels of intensity applying changes to the gender feature.

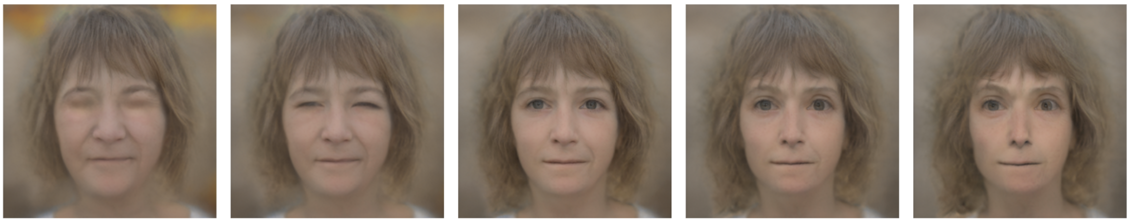


Figure 7.14: Different levels of intensity applying changes to the ’eyes open’ feature.

Figures 7.12, 7.13 and 7.14 show what the "uncanny valley" theory means [5], and the importance in generating faces as realistic as possible, to avoid the unsettling or creepy look artificial faces can have.

8 Demonstration to *El Laboratorio de Sueño y Memoria*

A presentation was made for Cecilia Forcato, the director of *El Laboratorio de Sueño y Memoria*, to show her the work done and the limits encountered. The idea was also to receive feedback from Dra. Cecilia Forcato.

The presentation began with a live demonstration of **FG-Style** done via ssh connection to the computer hosting both the front end and back end applications at ITBA. Graphical interfaces were used in order to display the front end and the images generated. Figures 6.3 and 6.4 show the different windows that were used to show the program to Cecilia. The parameters used to generate transitions (first and second face id, distance from first face and amount) were explained using a visual diagram to clarify any doubts. Figure 8.1 shows the diagram used. Later, results which can be seen in Section 7 were shown and discussed with Cecilia.

We told Cecilia that we knew since the beginning of the project that what we wanted to do was to keep the frame of the face and only change its center. We also wanted to avoid smiling, avoid glasses, and avoid all of the things that did not serve the Lab. What ended up happening is that we found out that the network does the opposite. It first changes the hair, the background, the orientation of the face, and all of the attributes that consume more resolution in the photo. It lastly changes the finest attributes that would be the nose, eyes, mouth, etc.

We explained to Cecilia that when we tried to change a specific face attribute (a smile for example), the network threw up results where the face lost its credibility. The attribute was changed, but the face didn't look human anymore. So when we tried to get in the network to change things that Cecilia had previously told us that she needed, realism was lost. The workaround we found to attribute modification and to generate similar faces was to make transitions. If a face has a smile and Cecilia wants it to be serious, a transition should be made to a face which is serious in order to modify that attribute. The same example applies to a face with glasses. Consequently, you can keep or change certain attributes if you make the transition right. This is the workaround we found to avoid the difficulty in modification of the faces generated by GANs. So if the faces to transition are well chosen, the faces from and to, she can play with which attributes are being changed or maintained.

Another remark made to Cecilia in the presentation was that the network used was trained with a specific dataset: the original. Future implementations could try to train the network with another dataset and other results could be obtained. For instance, if the dataset has no photos with smiles, the GAN won't create faces with smiles because it didn't learn how to do that while being trained.

Cecilia made a series of observations during the presentation. The main feedback was regarding the whole use of the project. She believes that **FG-Style** is good and useful, and that the scientific question that one asks oneself depends on the faces chosen. This means that **FG-Style** can be a tool for new future experiments and not only for the mock crime with lineups experiment which was the starting point of this project. Cecilia said that they could start exploring new ways in which people can create false memories like choosing the wrong criminal although the contour in faces is different, or they could use the transitions between two faces to explore new things. So instead of having an experiment and trying to use the tool for that experiment, she thought: "I have a tool that does this. Now lets think the experiment backwards. The experiment with this tool."

She pointed out that if she wanted to use the tool for the specific experiment of the mock crime with lineups, it would be useful to modify it so that she doesn't have to generate a huge amount of random faces till she finds the faces with the attributes she needs. She said that it would be of real use that she could ask the tool to generate the faces with the attributes she needs (e.g. a man with a beard). We told Cecilia that this was exactly something that we tried while experimenting with the GAN but it wasn't possible, mainly because results became non realistic. Yet, she insisted that when preparing a lineup in an experiment, certain characteristics are previously chosen between the scientists and it is really tedious to look for those attributes in millions of random faces. We thought we could bypass this problem by generating an additional tool that would look over all of the random faces generated, analyzing them and looking for certain characteristics that Cecilia needs. So with this tool you can generate a huge amount of random faces and then narrow that amount using this filter tool. Cecilia thought this could work really well.

Cecilia then asked if she could bring images of faces, add them to the program and generate transitions. This is a great and very useful idea, the problem is that this is not possible to do with StyleGAN. In fact, this is a great problem at a broader level of the discipline. When you do that, you lose the strength that the network has to generate faces that are realistic and similar to the originals. She also asked if faces in the middle of a transition could be used to transition to other faces. Another great idea she suggested, but not possible to carry out.

The demonstration ended with the previous discussion and we could sense that Cecilia had really liked the tool but that it wasn´t the optimal tool for the mock crime with lineups experiment they are working on because they have to do some manual work to find the ideal faces for the experiment. Future iterations could find solutions to simplify the work for the lab.

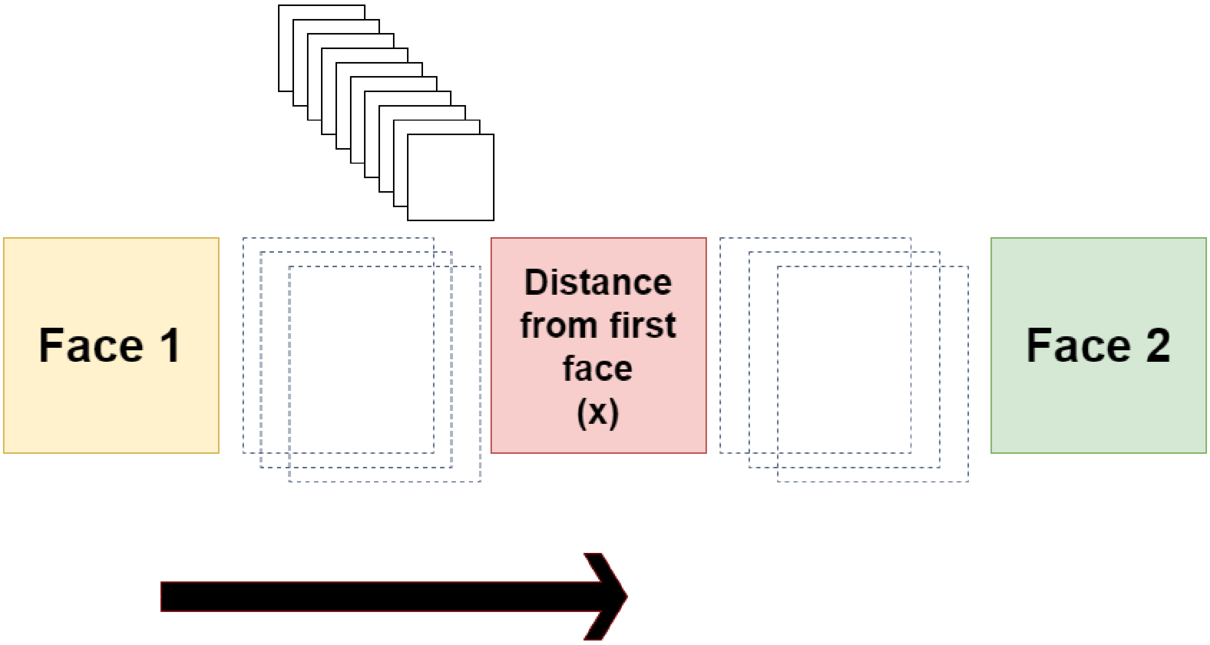


Figure 8.1: Parameters used to generate transitions. Note that the "distance from first face" red square can move from Face 1 till Face 2.

9 Future Implementations

There are plenty of functionalities and improvements that could be included to add value to the product and also to the investigation that we have today. Some of the ideas proposed to be added in the future are the following:

9.1 Filtering tool

In order to find faces with specific attributes needed for the creation of lineups in the experiments, random faces have to be generated till those attributes are found. **FG-Style** provides great variety but this comes with a drawback: a large amount of images has to be generated. Once this is done, someone should manually check the generated faces till the ones that match the attributes they want are found.

What we are proposing as a future feature is to create a tool that analyzes images and looks for certain specified attributes in order to filter the random faces generated with **FG-Style**. In order to filter, algorithms that detect objects or image segmentation algorithms which segment things in a photo could be used. *YOLO Algorithm for Object Detection* [42] is an example of an algorithm which gives real-time object detection with the use of neural networks. Article [1] proposes another solution for real-time attribute detection using deep learning. This idea could be transferred to the filtering tool, with the difference that it would not be used with real-time face captures.

This will simplify the Lab's job to get faces with the attributes they want.

9.2 Deployment, security and user experience

Currently, both applications are not deployed, they can only be used locally, by using the computer at ITBA or using ssh. The deployment of the **FG-Style** server was attempted but failed as the users ITBA provides us are LXC containers and the deployment could not be done through containers. Moreover, the CUDA compiler (nvcc) aroused errors when trying to deploy the application using service monitors for the Gunicorn's deamon mode running application such as Supervisor, Runit and Systemd, which do not occur when running Gunicorn manually and normally. We believe that with more time, these problems could be overcome and the deployment of the applications should be implemented in the future. Both the **FG-Style** app and the **FG-Style** server could be made available to use from other locations, with the use of identification. Therefore, identification implementation should be also added to the solution.

Security is an attribute we did not prioritize when thinking about the solution design. It should definitely be taken into account when deploying the applications, and the REST API of the generator should be revised in order to make the endpoints as secure as possible. Gunicorn offers features for production applications that we are currently not taking advantage of, such as having more than two workers to listen and handle the HTTP requests in parallel, and using SSL. Also, error handling should be added to the API, for the users to have a better understanding of the errors.

Finally, user experience can be improved. Because the applications run locally, results are saved in a local folder, but they could eventually be shown in the front-end application by having the results be persisted in a cloud service such as AWS S3 by the back-end application, and the front-end can consume that service to consult results, becoming independent of the local folder the back-end application currently uses.

9.3 Find new ways to modify attributes

Future implementations could find new alternatives to move within the latent space in a meaningful way to be able to modify attributes. Figuring out what one is supposed to modify of the latent vector in order to edit specific attributes of an image in a consistent way is really difficult. “Taking a small step in a random direction will most likely change more than one aspect of the photo since latent spaces of most well-known generators are rather entangled, meaning that by adding a smile to the generated face you are likely to also unintentionally change the hair color, the eye shape or any number of other wacky things” [2]. In the paper [47], researchers propose a partial solution to change certain attributes such as age, gender and background removal. Studying and working with this paper could further improve the existing tool **FG-Style**.

9.4 Custom datasets

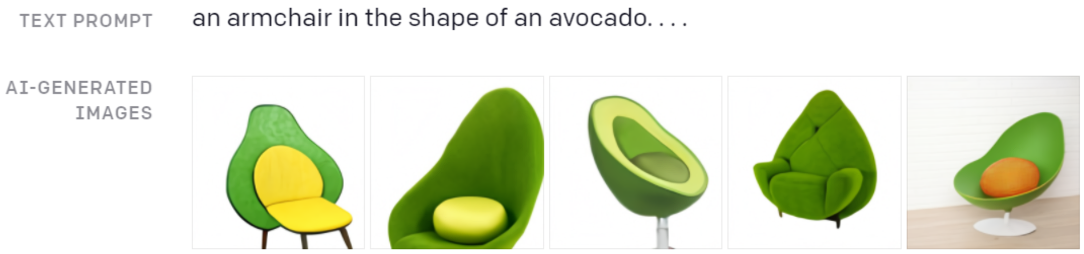
Training the network with curated datasets will make the faces created be more like what the lab needs (e.g. if no face in the training dataset has a smile, no random face with a smile will be generated). This could also be useful to experiment with different ethnicities: a dataset of latin american faces could be used so that StyleGAN generates latin-american-like faces. In fact, in the Appendix we talk about the collaboration done with PhD Pablo Negri and his team at the Image Processing and Computer Vision Group that belongs to the Imaging and Robotics area of *El Instituto de Ciencias de la Computación* (ICC) [4]. In this collaboration, a dataset of latin american faces taken from IMDB's database is being done. This dataset could be used to train the network.

It is important to note that training StyleGAN requires a high computational effort so it has to be done with a powerful GPU. Google Colab also allows training but in the cloud. Article [40] explains how StyleGAN can be trained step by step.

9.5 New technologies

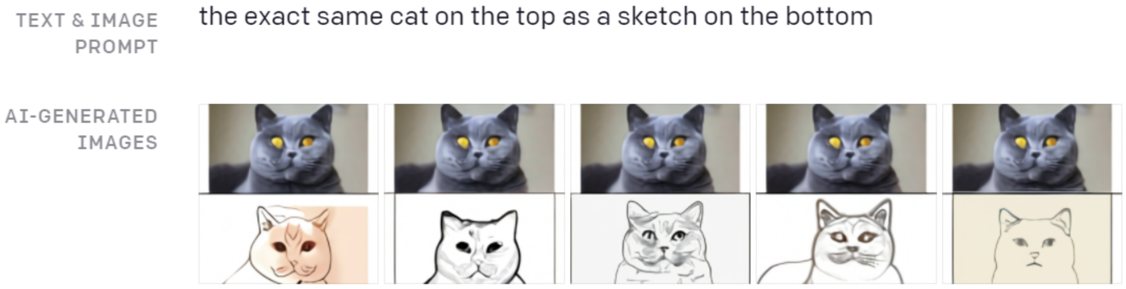
Other GANs or technologies could be explored in order to find other ways to generate random faces and manipulate them in a way that is suitable for the Lab. One particular new technology that we found interesting to delve into as a future implementation is OpenAI's Dall-E creation.

Dall-E [39] [6] is a 12-billion parameter version of GPT-3 [7] that creates diverse styled images from textual descriptions. Dall-E can create images of objects which are realistic ("a store front that has the word 'openai' written on it") along with images of things that do not exist ("a giraffe with the body of a turtle"). Figures 9.1 and 9.2 illustrate what Dall-E does.



an armchair in the shape of an avocado. . . .

Figure 9.1: Dall-E example



TEXT & IMAGE PROMPT

the exact same cat on the top as a sketch on the bottom

AI-GENERATED IMAGES

Figure 9.2: Dall-E example

Dall-E could be really useful to generate random faces with specific attributes. So for example, people in the Lab could ask the program to generate "the face of a 50 year old man, with a beared, eyes apart, white skin and red hair".

10 Conclusion

During this project we have investigated various applications of Artificial Intelligence and the amazing possibilities they have to be of great help in problems that for humans may seem very simple, such as facial recognition. During the particular study of a GAN implementation like StyleGAN and all the GANs that are included in the state of the art of this field in AI, we were able to appreciate and understand how AI grew over the years and how technological advances related to hardware greatly accompanied and boosted this advance and all the uses that emerged in a little over 10 years to make incredible Deep Learning applications. We are convinced that the future of this field will be even more exciting.

The study of neurosciences and the formation and alteration of memory allowed us a deep understanding of the processes that occurs in our mind and gives us our identity. We learned that unfortunately our society is not properly prepared for it, since the Innocence Project is a clear demonstration of how there are flaws in judicial systems that can impact a person's life, and that the work they do with *El Laboratorio de Sueño y Memoria* at ITBA gives us hope that society is also trying to correct itself. If we can make a small positive contribution to this cause, or if this project also inspires others to help even more by improving the project or starting others, our work is done.

On the other hand, the StyleGAN implementation also gave us a lot of resilience to meet all hardware and software requirements, and finally achieve an application that could interact and use the pre-trained network. We still have to test a network trained by our own dataset in the future, but it was not within the scope of this project, since we consider that it is an investigation outside the implementation of the network and generation of potentially usable results for *El Laboratorio de Sueño y Memoria.* Regarding the implementation of the applications, it was difficult for us to meet our own usability requirements, to make the applications available and show the results constantly, but they have performed very well computationally, because by using everything locally, the network can quickly generate hundreds of images in minutes. We also have the peace of mind that these applications are made in such a way that improvements can be built on them in future projects.

Regarding the results, we observed that StyleGAN is a tool that meets and exceeds what we had previously investigated, since the level of resolution and quality of the facial images left us astonished. In addition, computationally the network generates the images with great speed, something that allowed us to use the network many times. Using the FFHQ dataset for the pre-trained network, we observed that many faces contained accessories, stands or poses, smiles, and attributes that did not serve us in terms of generating random faces, but we were able to adapt the application to manually filter the faces with which you want to work and study.

Modifying the most central and important features, while maintaining the least important ones, in the structure of a face was really a challenge. StyleGAN is trained and designed in such a way that the attributes most sensitive to changes are the least important in terms of the structure of the face, the previously mentioned low-level features, and the most difficult to modify are the attributes that most needed to change in laboratory studies, the ones that most identify and form one's face. As the network constantly changed the face frame in the transitions, and the center changed with less aggression, we had a great contradiction with what the laboratory asked us. That is why we made the decision to investigate latent space in order to obtain the latent vector of an encoded face and make modifications to specific high-level features. It can be seen from the results that this experimentation did not obtain faces that appear to be realistic enough and with good enough quality to be used in human studies. However, it is also for future implementations to carry out further research on this projection, since the features we wanted to change seem to be the correct ones, giving us a clue that the latent directions we obtained could be useful given a better projection. Finally, we made the decision to make greater use of the style mixing properties that identify StyleGAN and to offer this as the possibility for the laboratory to generate faces similar to a selected one. As a disadvantage, users will have to manually select the faces (or styles) to mix and manually select the input parameters to generate a transition with similar faces that maintain the same attributes within the face frame and differ in high-level features. It will require many uses to become familiar with the application, but in the results we observed that even when making transitions between very different styles, the possibility of controlling how far to go in the transition was very useful so as not to end up generating faces that were too different and useless for experiments.

It is definitely interesting how the StyleGAN network does what a human brain mentally processes when learning a face: the central structure of the face is the most important, but the frame of the face is what identifies it the most. We believe that if you manually select which styles to mix, and with how much distance and amount to transition, you can find useful results for the laboratory, where faces manage to be alike, but different, and meet the aesthetic requirements of the laboratory for experiments. We hope that this triggers among our more experienced readers the need to study this relationship in more depth, to understand how the similarities and differences between Deep Learning and the human mind can help experiments in laboratories, or even modify them.

11 Acknowledgements

Our greatest appreciation goes to our tutor, Dr. Rodrigo Ramele, who guided us along the realization of the project with a lot of encouragement, enthusiasm and support, giving us all the tools and resources necessary to complete the project.

We are extremely grateful for Instituto Tecnológico de Buenos Aires and the IT team that helped us set up our work at the Titan Xp GPU, which was provided by the University for our work. We would also like to thank Dr. Cecilia Forcato and her team in the Laboratorio de Sueño y Memoria, who were of great assistance at explaining us how the experiments are conducted, were available for us to further understand how this project could be implemented for their use and fulfill its purpose, and give us feedback of our progress with the application and its design.

It was a great privilege to work with PhD Pablo Negri and his team at the Image Processing and Computer Vision Group that belongs to the Imaging and Robotics area of *El Instituto de Ciencias de la Computación* (ICC) [4].

Finally, this project could not have been done without the continuing support of our family, friends and loved ones.

References

[1] Real-time multi-facial attribute detection using transfer learning and haar cascades with fastai. https://towardsdatascience.com/real-time-multi-facial-attribute-detection-using-transfer-learning-and-haar-cascades-with-fastai-47ff59e36 12 Dec., 2011 (accessed October 1, 2021).

[2] Unsupervised discovery of interpretable directions in the gan latent space (5-minute summary). https://www.reddit.com/r/MachineLearning/comments/q38a35/d_paper_explained_unsupervised_discovery_of/, 07 Oct., 2021 (accessed October 10, 2021).

[3] https://exactas.uba.ar/, (accessed October 1, 2021).

[4] https://www-2.dc.uba.ar/grupinv/imagenes/test/index.php/Home/, (accessed October 1, 2021).

[5] https://spectrum.ieee.org/what-is-the-uncanny-valley, (accessed October 1, 2021).

[6] https://openai.com/blog/dall-e/, (accessed October 1, 2021).

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[8] Philip Dale, Elizabeth Loftus, and Linda Rathbun. The influence of the form of the question on the eyewitness testimony of preschool children. *Journal of Psycholinguistic Research*, 7:269–277, 07 1978.

[9] NVIDIA developers. Stylegan2 — encoder/projector for official tensorflow implementation. https://github.com/rolux/stylegan2encoder, 2019 (accessed October 10, 2021).

[10] Patrick Esser, Robin Rombach, and Björn Ommer. A note on data biases in generative models. https://arxiv.org/abs/2012.02516, 2020 (accessed October 10, 2021).

[11] Cecilia Forcato. *Estudio de la fase de reconsolidación de la memoria declarativa en humanos.* PhD thesis, 2011.

[12] Cecilia Forcato, Pablo Argibay, María Pedreira, and H Maldonado. Human reconsolidation does not always occur when a memory is retrieved: The relevance of the reminder structure. *Neurobiology of learning and memory*, 91:50–7, 11 2008.

[13] Cecilia Forcato, Rodrigo Fernandez, and María Pedreira. Strengthening a consolidated memory: The key role of the reconsolidation process. *Journal of physiology, Paris*, 108, 09 2014.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016 (accessed October 1, 2021). http://www.deeplearningbook.org.

[15] Rani Horev. Explained: A style-based generator architecture for gans - generating and tuning realistic artificial faces. https://towardsdatascience.com/

explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431, 2018 (accessed October 1, 2021).

[16] Ira Hyman, Troy Husband, and F. Billings. False memories of childhood experiences. *Applied Cognitive Psychology*, 9:181 – 197, 06 1995.

[17] Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville Yoshua Bengio Ian J. Goodfellow, Jean Pouget-Abadie. Generative adversarial nets. https://arxiv.org/abs/1406.2661, 2014 (accessed October 1, 2021).

[18] Women in Data. Women in data. https://www.womenindata.org/, (accessed September 5, 2021).

[19] Eric R. Kandel. The Molecular Biology of Memory Storage: A Dialog Between Genes and Synapses. *Bioscience Reports*, 24(4-5):475–522, 08 2005.

[20] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors. *Principles of Neural Science*. The McGraw-Hill Companies, Inc, New York, fifth edition, 2000.

[21] Animesh Karnewar and Oliver Wangr. Multi-scale gradients for generative adversarial networks. https://arxiv.org/abs/1903.06048, 2019 (accessed October 1, 2021).

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. https://arxiv.org/abs/1710.10196, 2017 (accessed October 1, 2021).

[23] Tero Karras and Janne Hellsten. Ffhq dataset repository. https://github.com/NVlabs/ffhq-dataset, 2019 (accessed October 10, 2021).

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. https://arxiv.org/pdf/1812.04948.pdf, 2018 (accessed October 1, 2021).

[25] Tero Karras, Samuli Laine, Timo Aila, Miika Aittala, Janne Hellsten, and Jaakko Lehtinen. Analyzing and improving the image quality of stylegan. https://arxiv.org/pdf/1912.04958.pdf, 2019 (accessed October 1, 2021).

[26] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. https://arxiv.org/abs/1906.02691, 2019 (accessed October 1, 2021).

[27] Elizabeth Loftus. Make-believe memories. *The American psychologist*, 58:867–73, 12 2003.

[28] Elizabeth F. Loftus. Creating false memories. *Scientific American*, 277(3):70–75, 1997.

[29] Elizabeth F Loftus and Jacqueline E. Pickrell. The formation of false memories. *Psychiatric Annals*, 25:720–725, 1995.

[30] Jimena Lozano and Maite Herrán. Transition results from face 1 to face 2. https://youtu.be/MLxHHgP2Vdg, 2021 (accessed October 10, 2021).

[31] Jimena Lozano and Maite Herrán. Transition results from face 1 to face 3. https://youtu.be/V3VcWPGhKK0, 2021 (accessed October 10, 2021).

[32] Jimena Lozano and Maite Herrán. Transition results from face 1 to face 4. https://youtu.be/LuHGlNsWVrI, 2021 (accessed October 10, 2021).

[33] Kathleen McDermott and Jason Watson. The rise and fall of false recall: The impact of presentation duration. *Journal of Memory and Language*, 45:160–176, 07 2001.

[34] James Mcgaugh. Memory-a century of consolidation. *Science (New York, N.Y.)*, 287:248–51, 02 2000.

[35] Peter Norvig and Stuart Russell. *Artificial Intelligence: A Modern Approach.* Prentice Hall Press Upper Saddle River, 2009.

[36] Innocence Project. How many innocent people are in prison? https://innocenceproject.org/how-many-innocent-people-are-in-prison/, 2011 (accessed September 5, 2021).

[37] Innocence Project. In focus: Eyewitness misidentification. https://innocenceproject.org/in-focus-eyewitness-misidentification/, (accessed September 5, 2021).

[38] Innocence Project. Research resources. https://innocenceproject.org/research-resources/, (accessed September 5, 2021).

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021.

[40] Fathy Rashad. How to train stylegan2-ada with custom dataset. https://towardsdatascience.com/how-to-train-stylegan2-ada-with-custom-dataset-dc268ff70544, 2016 (accessed October 1, 2021).

[41] Sexta Redacción. El itba lanza laboratorio de sueño y memoria. https://www.sextaseccion.com/2020/03/29/el-itba-lanza-laboratorio-de-sueno-y-memoria/, 07 Apr., 2020 (accessed September 5, 2021).

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[43] Henry Roediger and Kathleen McDermott. Creating false memories: Remembering words not presented in lists. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21:803–814, 07 1995.

[44] Juergen Schmidhuber. Deep learning in neural networks: An overview. https://arxiv.org/abs/1404.7828, 2014 (accessed October 1, 2021).

[45] Christopher Schmidt. Training at home, and in the cloud. https://www.chrisplaysgames.com/gadgets/2019/02/26/training-at-home-and-in-the-cloud/, 2019 (accessed September 5, 2021).

[46] Alyssa H. Sinclair and Morgan D. Barense. Prediction error and memory reactivation: How incomplete reminders drive reconsolidation. *Trends in Neurosciences*, 42(10):727–739, 2019.

[47] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In Hal Daumé III and Aarti Singh, editors,

Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 9786–9796. PMLR, 13–18 Jul 2020.

[48] Kimberley Wade, Maryanne Garry, John Read, and D Lindsay. A picture is worth a thousand lies: Using false photographs to create false childhood memories. *Psychonomic bulletin review*, 9:597–603, 10 2002.

[49] John T. Wixted and Gary L. Wells. The relationship between eyewitness confidence and identification accuracy: A new synthesis. *Psychological Science in the Public Interest*, 18(1):10–65, 2017. PMID: 28395650.

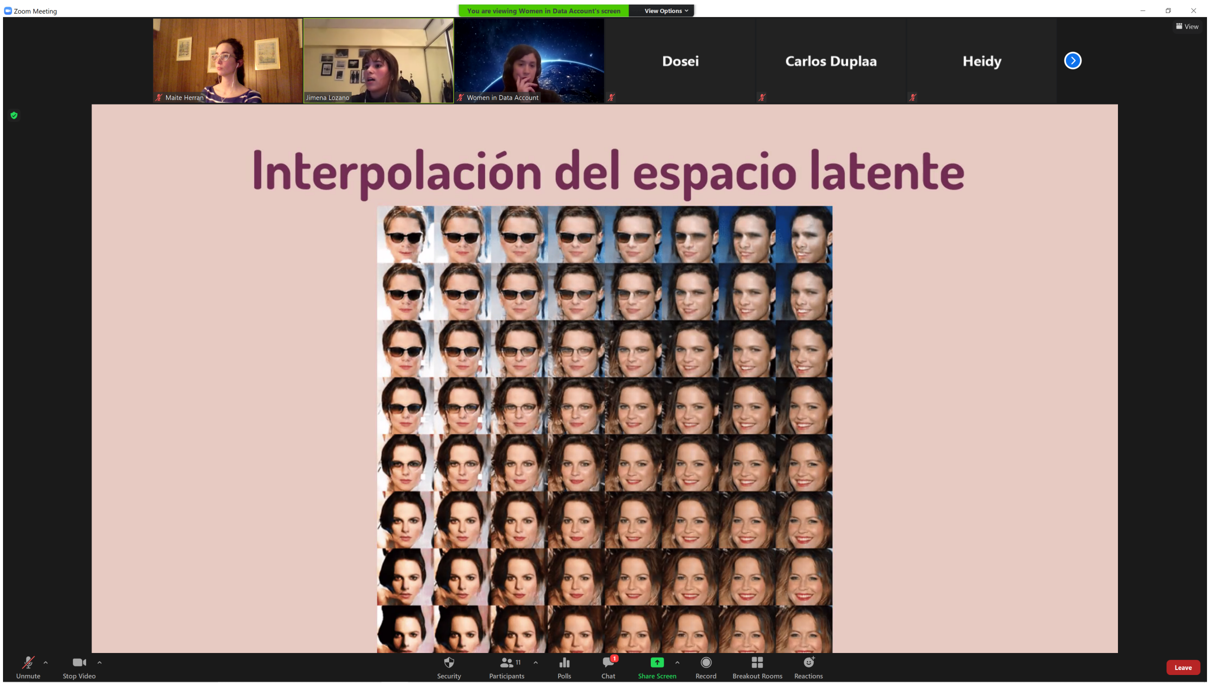
12 Appendix

12.1 Women In Data

Women in Data is a non profit organization whose mission is to "close the gender gap and increase diversity in data careers" [18]. The organization was founded by Sadie St Lawrence in 2014, after noting the limited number of women in her masters program in data science. Sadie felt uneasy upon "the prospect of gender equality as she knew the future of tech and business were all data driven" [18]. She then started Women in Data in 2015, a community to incorporate diversity in data careers.

Soledad Álvarez del Sel is Women in Data's Buenos Aires's Chapter Lead. Soledad contacted us so that we could be the first speakers to deliver an online talk in the Chapter held in Buenos Aires. The talk was done on June 23rd 2021 via a zoom meeting in front of an audience.

We toured the audience through memory, false memories, machine learning, deep learning, GANs and StyleGAN. It was a one hour speech where we tried to condense almost everything we covered in this work and which ended with a space of exchange of questions and answers.



You are viewing Women in Data Account's screen

Interpolación del espacio latente

Figure 12.1: Screenshot from the Women in Data talk

12.2 Collaboration with Pablo Negri

Dr. Pablo Negri is in the Image Processing and Computer Vision Group of the Computing Department of the FCEyN-UBA [3]. It is one of the groups that belong to the Imaging and Robotics area of *El Instituto de Ciencias de la Computación* (ICC) [4]. Under Pablo's direction is Camila Di Ielsi, who is working in the creation of a dataset of latin-american-like faces using IMDB's database. This database could be used in the future to train the network. The Lab will therefore be able to generate faces with different ethnicities and study if there is any bias in eyewitness criminal identifications.

It is important to stress that the database Camila is working on has faces which are smiling. If the StyleGAN network is to be trained with this dataset, this should be taken in to account. If no smiling faces are wanted to be generated, this faces should be removed.