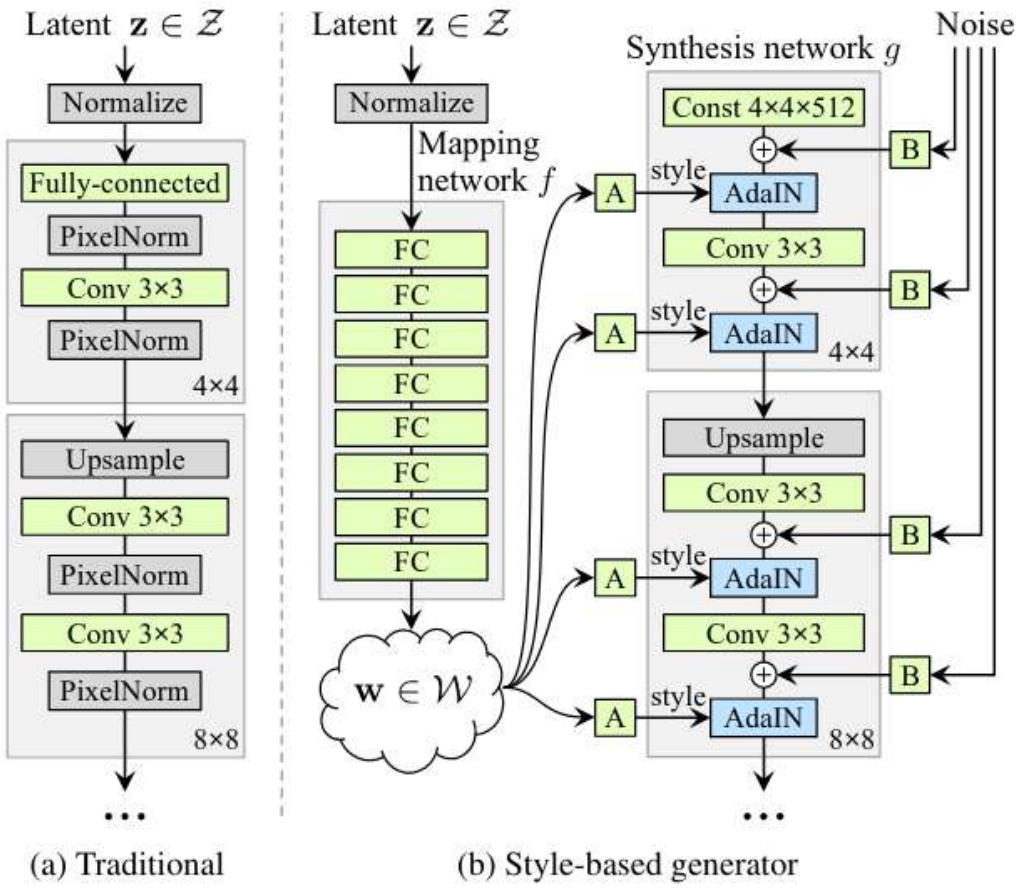
Exploration of StyleGan Capabilities and Limits

Brayam Castillo, Gabriel Baker, Filipe Lauar, Vinícius Imaizumi

Introduction

Recently, the results of generative adversarial networks have rapidly improved, being capable of generating convincely enough images of faces that do not exist, but that can trick humans into believing them. A neural architecture that has made a big step in this direction is StyleGan [2], which not only generates real looking faces, but also allows for meaningful displacements in the latent space, allowing for the change of specific attributes.

The architecture of the StyleGan is similar to previous works, but it has 3 main differences, as shown in the image below, in order to better model the latent space, with a separation of high level attributes and stochastic variation in the synthesized image. The first one is a remapping of the latent space into an intermediate latent space. The second difference is the passage of the latent code to layers of the synthesis network, that uses an affine transformation to extract styles, this is then mixed with the imaging passing through the network with the AdaIn function. And the last change is the introduction of new noise after every convolution, giving the model a source of variation to characteristics such as hair and freckles.



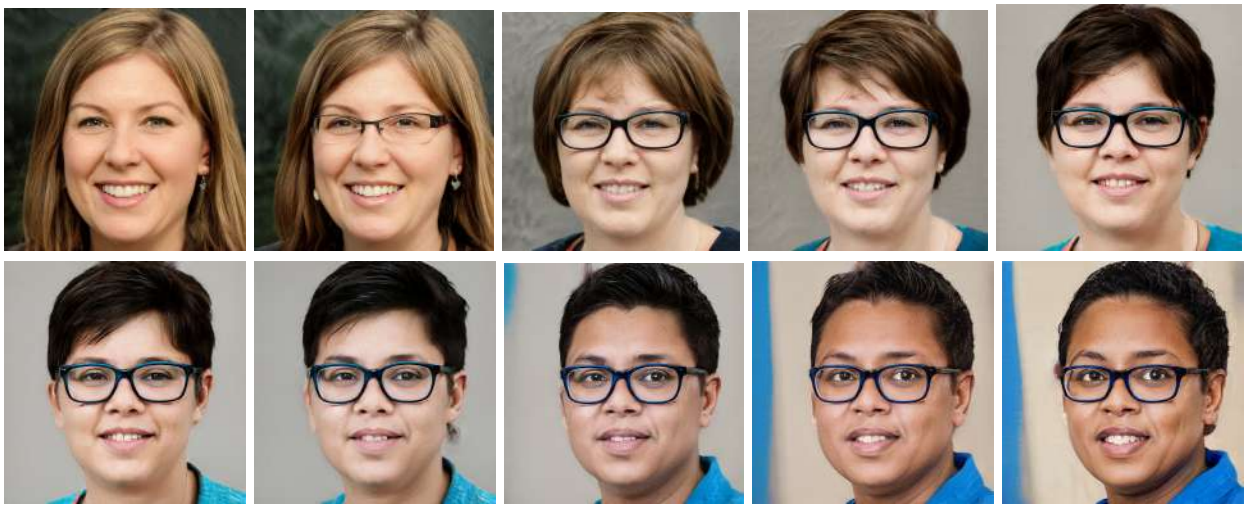
In this work we have analysed the capabilities of this network, first testing the results of a walk between two people and later testing the variation of a single feature, as well as its impact on other characteristics of the person. For the latter to be possible, we used the InterFaceGAN [3] architecture to find the direction of variation of a single attribute and a neural network to automatically extract features [3] in order to check for unwanted changes. We have also worked with images of real people, converting the photo to the latent space, performing the change, and bringing it back into an image.

Weighted average in latent space between two images

Since the styles transformations applied in StyleGan are directly related to the image in the latent space $\mathscr{W}$, we can manipulate the images in this space to change its style. A very simple way to do this type of transformations is to do the weighted average between two images I and J in the latent space $\mathscr{W}$:

$$T(I, J, k) = kI + (1 - k)J$$

In our case, we did this weighted average with $k \in \{0, 1/9, 2/9, ..., 8/9, 1\}$. Thus, in the extremes we have $T(I, J, 0) = J$ and $T(I, J, 1) = I$. We can see the results in the images below generated with the pretrained network of the github paper repository[1].



Transitions between images J (upper left) and I (bottom right).

Visually we achieve a very good result. It is interesting to observe that every single one of these images indeed represent a possible person. The feature transition is very smooth. We can see gradually a transition for the glasses from none, to a glass with a weak eyeglass stem, to a black eyeglass and finally a dark-blue color at the end. Similar transitions happen for the hair, both in color and type, for the skin color and even for the T-shirt, for its color and its presence, which might be related to the pose in the image.

[1] https://github.com/NVlabs/stylegan

This experiment shows the StyleGAN was able to perform the style mixing.

Moving in the latent space W to change a specific attribute

A more sophisticated way to manipulate a image in latent space $\mathscr{W}$ is to directly change the values of its coordinates in this space. However, the directions orthogonal to the base vectors don't necessarily have a human interpretation. That is, we do not know beforehand what are the effects of moving through the first coordinate in terms of perceptible styles, nor if moving through this space generates a person-like image. For instance, if we change the 47th dimension of the image I from -0.42 to 20, we have the result below (left) and from -0.42 to 40 we have the result below, on the right.

Although we can still recognize the face, it still has an artifact and going from 20 to 40 the artifact got even worse. Also, the person generated has different characteristics from the person in the image I.

In order to find a good direction to move in the latent space to change a specific characteristic and keep as much as possible the same person as before, that is, without changing uncorrelated attributes with the one we want to change, we used InterfaceGAN (InterfaceGAN).

It is possible to change 5 attributes using InterceGAN, namely, age, eyeglasses, gender, pose and smile. In the image below we have an example for the age attribute, starting at distance -4 from the boundary and going up to distance 4 from the boundary.

Although we can clearly see a lot of similarities among the images, we can also see that there are differences. For instance, in the upper left image the person seems to be much older than the person in the bottom right. This age difference seemed to be gradual between each image. We were able to visually identify a lot of correlated attributes when changing one of the five attributes using InterfaceGAN, which were changed with a high consistency across different images. For instance, the smile was correlated negatively with age, as well as age was correlated with glasses, as we can see in the example below.

This correlation was reported in the InterfaceGAN paper, since different semantics might be entangled. A possible solution to this problem is the conditional manipulation to change a given attribute1 without affecting a specific attribute2. It is done by moving in the direction of the original direction's projections into the plan of attribute2, thus the distance to the attribute2 plan will remain the same. If we want the same but with respect to more attributes (attribute2, attribute3, …) we can apply the same idea and subtract the projection from the primal direction onto the plane constructed by all conditioned directions.

The InterfaceGAN github repository provided some conditional boundaries, but they are available only for the $\mathcal{Z}$ space. For a future work, one could find these conditional boundaries in $\mathcal{W}$ and verify the results.

Face attributes

In order to better analyze the changes in facial attributes after the displacements in the latent space done in the last section, we used a pre trained[2] version of the neural network architecture [1], which, given an input image of a face, will output a binary array representing 40 predefined attributes listed below.

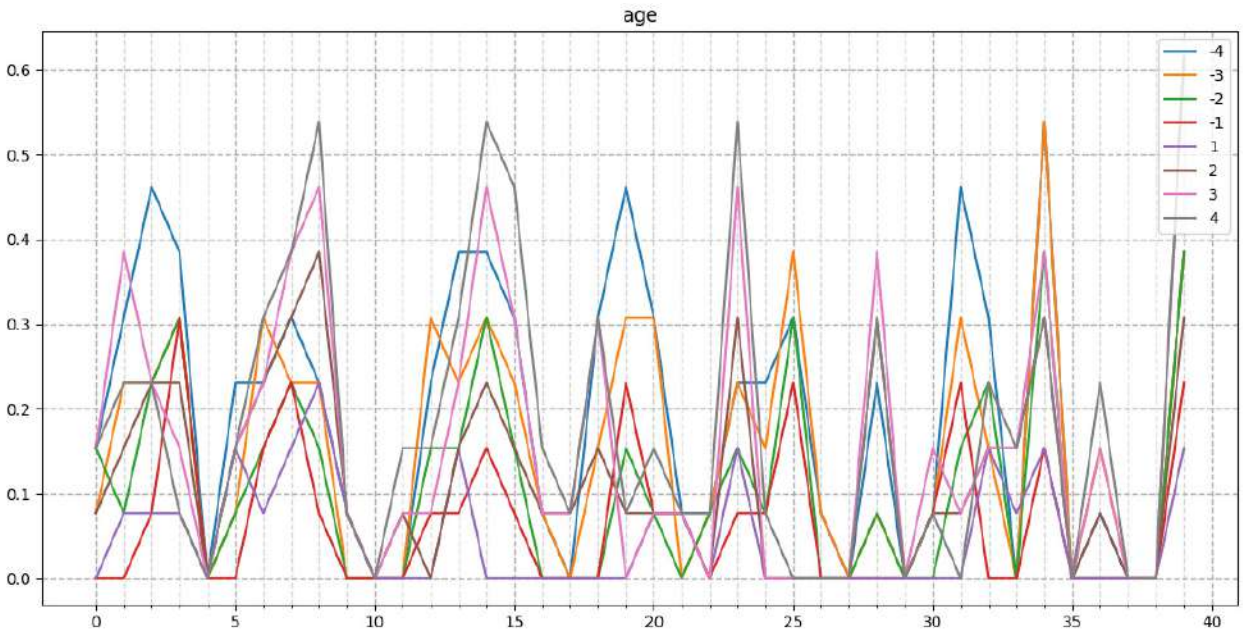
0	5_o_Clock_Shadow	10	Blurry	20	Male	30	Sideburns
1	Arched_Eyebrows	11	Brown_Hair	21	Mouth_Slightly_Open	31	Smiling
2	Attractive	12	Bushy_Eyebrows	22	Mustache	32	Straight_Hair
3	Bags_Under_Eyes	13	Chubby	23	Narrow_Eyes	33	Wavy_Hair
4	Bald	14	Double_Chin	24	No_Beard	34	Wearing_Earrings
5	Bangs	15	Eyeglasses	25	Oval_Face	35	Wearing_Hat
6	Big_Lips	16	Goatee	26	Pale_Skin	36	Wearing_Lipstick
7	Big_Nose	17	Gray_Hair	27	Pointy_Nose	37	Wearing_Necklace
8	Black_Hair	18	Heavy_Makeup	28	Receding_Hairline	38	Wearing_Necktie
9	Blond_Hair	19	High_Cheekbones	29	Rosy_Cheeks	39	Young

The usage of this network was intended to make the analysis of a large number of images easier, since it is orders of magnitude faster than a human evaluation, but it is less complete. It is also worth mentioning that the dataset used in training can introduce a bias in less objective

[2] https://github.com/TencentYoutuResearch/FaceAttribute-GAN

measures, for example in the attractive attribute, that is more likely to be false for people of color[3].

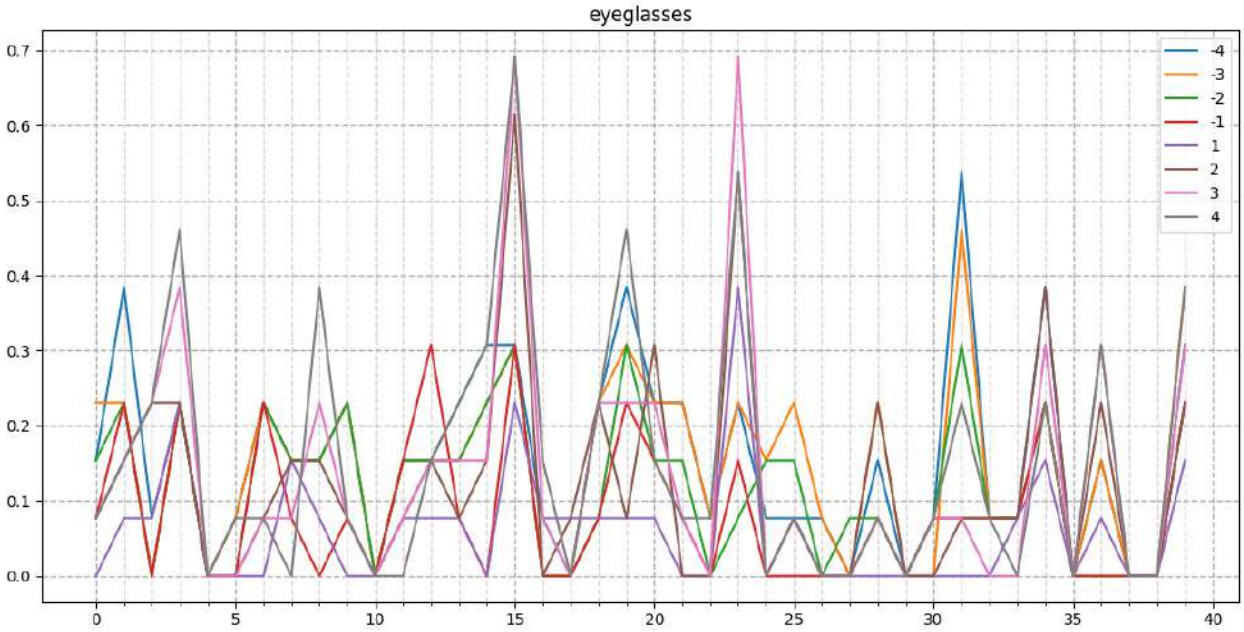
For the experiment we generated variations with a distance from a given boundary from -4 to 4 of 5 specific boundaries of 13 different base images, which had their features extracted. We then used these values to check the variation of features in relation to the respective base image (ie.: distance 0 to the boundary), and, for each of the 5 attributes we plotted a graph with the mean variation for each feature. The expected result was a sparse graph with variations only in relevant features, for example, when we changed the smile attribute it was reasonable for the high_cheekbones attribute to change and it is expected that the smiling and mouth_slightly_open change.



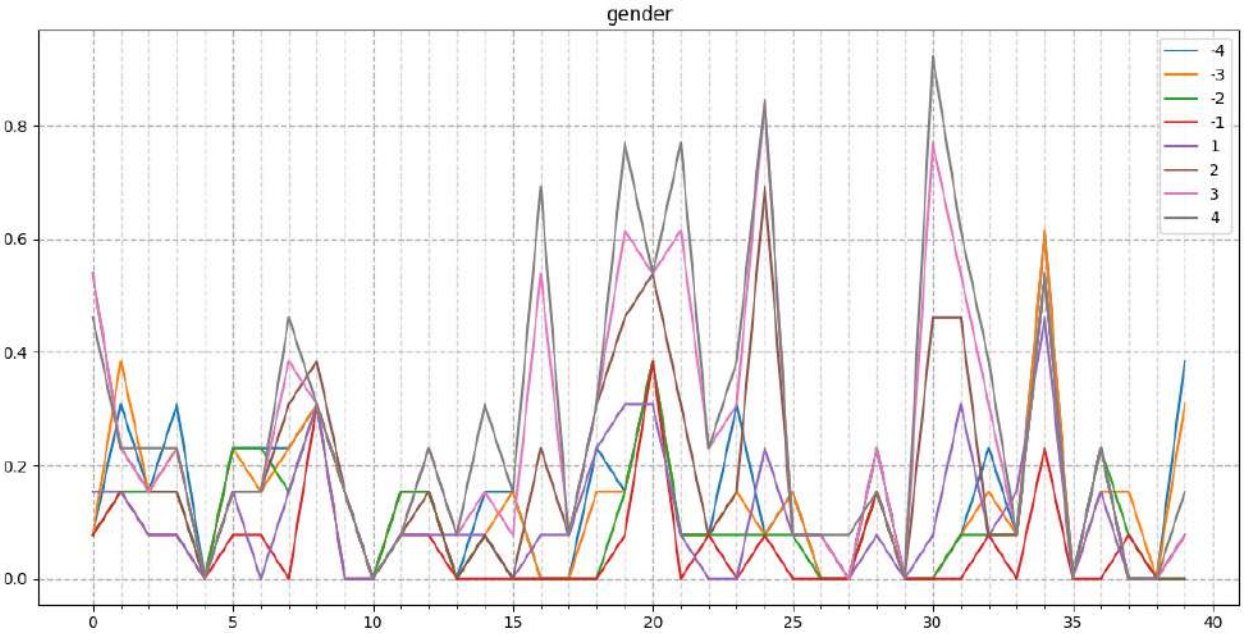
Mean variation of age with respect to distance 0 to the boundary.

Realistic modeling of age requires the model to take into account more changes in facial features and head shape. This is how, we can see that there are more attributes that drastically change like Wearing earrings, Narrow eyes and Double chin. Some others like Bushy Eyebrows or Waring Earrings remain unchanged as expected.

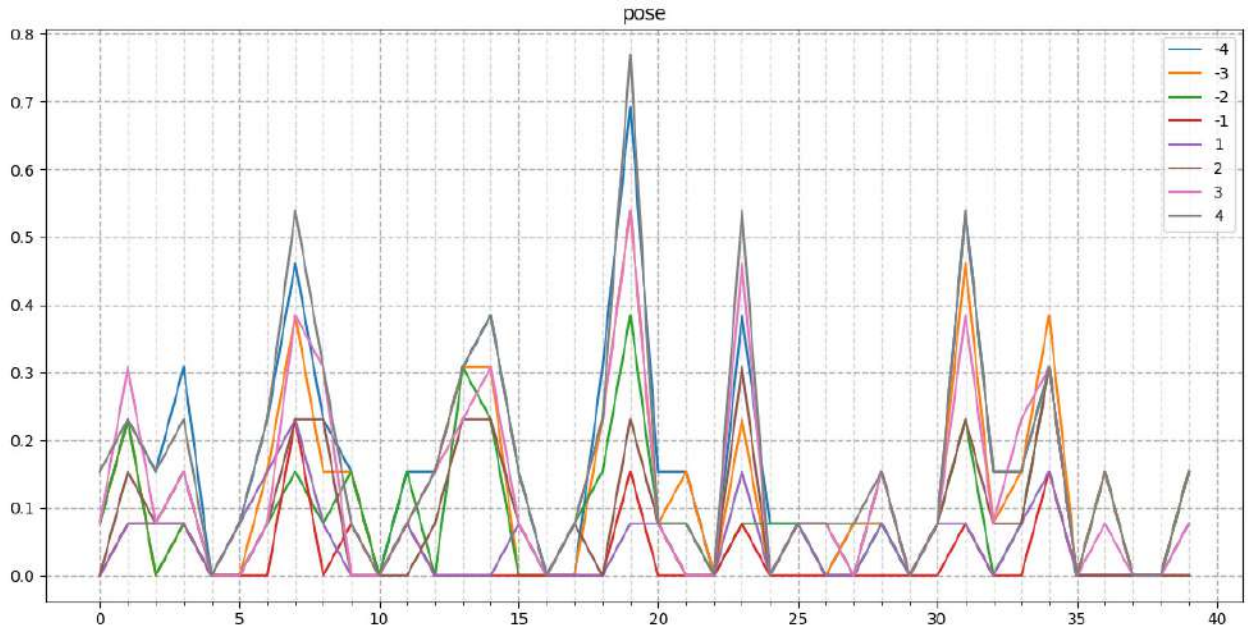
3 https://medium.com/@mike.leske/how-i-accidentally-created-a-racist-ai-by-a-naive-dataset-selection-2cc9bf369bfa



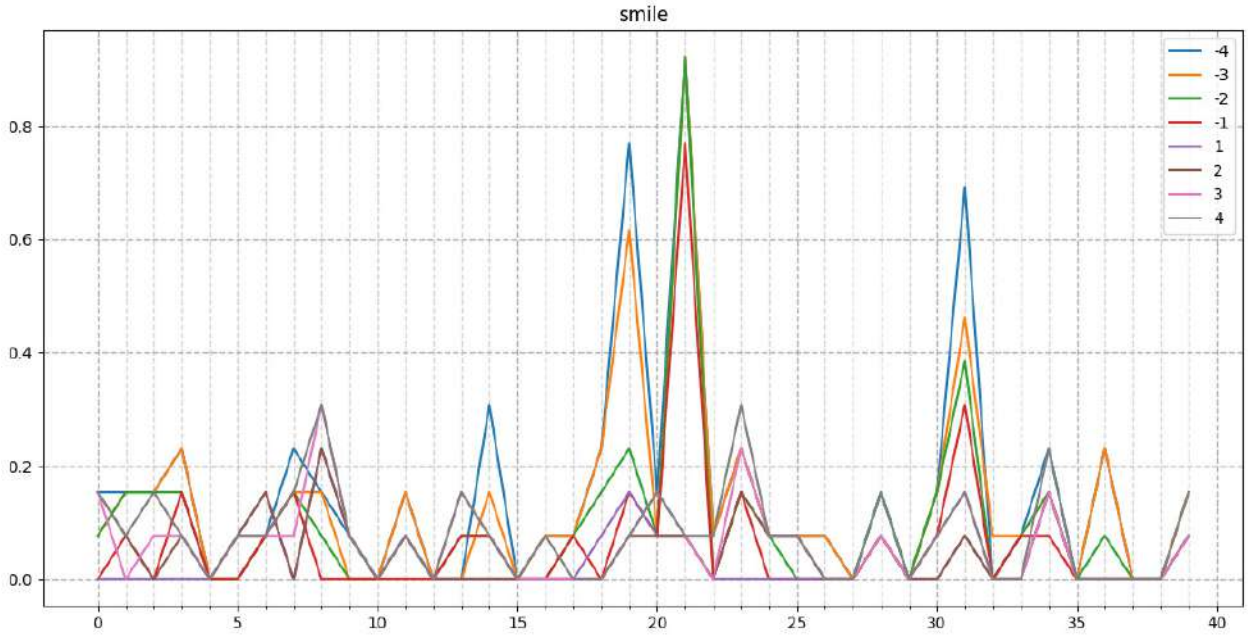
Mean variation of eyeglasses with respect to distance 0 to the boundary.



Mean variation of gender with respect to distance 0 to the boundary.



Mean variation of pose with respect to distance 0 to the boundary.



Mean variation of smile with respect to distance 0 to the boundary.

The features High Cheekbones, Mouth Slightly Open and Smiling change considerably when the method swap the attributes of the smile. In contrast, the attributes far from this region remain almost unchanged like the Hair or Eyeglasses, which was the expected result.

Overall we can see very different results, with a low amount of variation for the smile and a lot of variation for the age. Some variations are very logical, as in the smile case which

changed mainly the features High Cheekbones, Mouth Slightly Open and Smiling, while other variations might be due to a bias in the StyleGAN training set.

It is interesting to note that in all cases the attribute of index 4 - bald - remained the same. It could indicate that the dataset is biased, since the model might have trained with few images of bald people. It could also indicate that the variations in the 5 possible attributes using InterfaceGAN wasn't able to find a region in the latent space $\mathcal{W}$ related to baldness.

Image Inversion

To go further in applications of the style mixing and attribute changing, we used another network[4], named In-Domain Inversion GAN [4] to transform real people images to its latent code W. The approach to generate the latante code is quite simple, the researchers proposed a GAN architecture where the generator is an encoder decoder network that tries to, given a real image, recreate this exact real image. The discriminator aims to discriminate between a real and a generated image. Once the network is trained, we use the encoder part of the generator to transform the real image in its latent code W.

Having the latent code that generates the real image, we can perform the style mixing and also the attribute changing as well as we did before. The only difference between styleGAN and IDinverse is that the first uses images of size 1024x1024 and the second uses images of size 256x256, so the latent code W has also different sizes, 18x512 for the first and 14x512, so we can't mix images from both networks.

Below we can see in the left the real images and in the right the respective generated images.

4

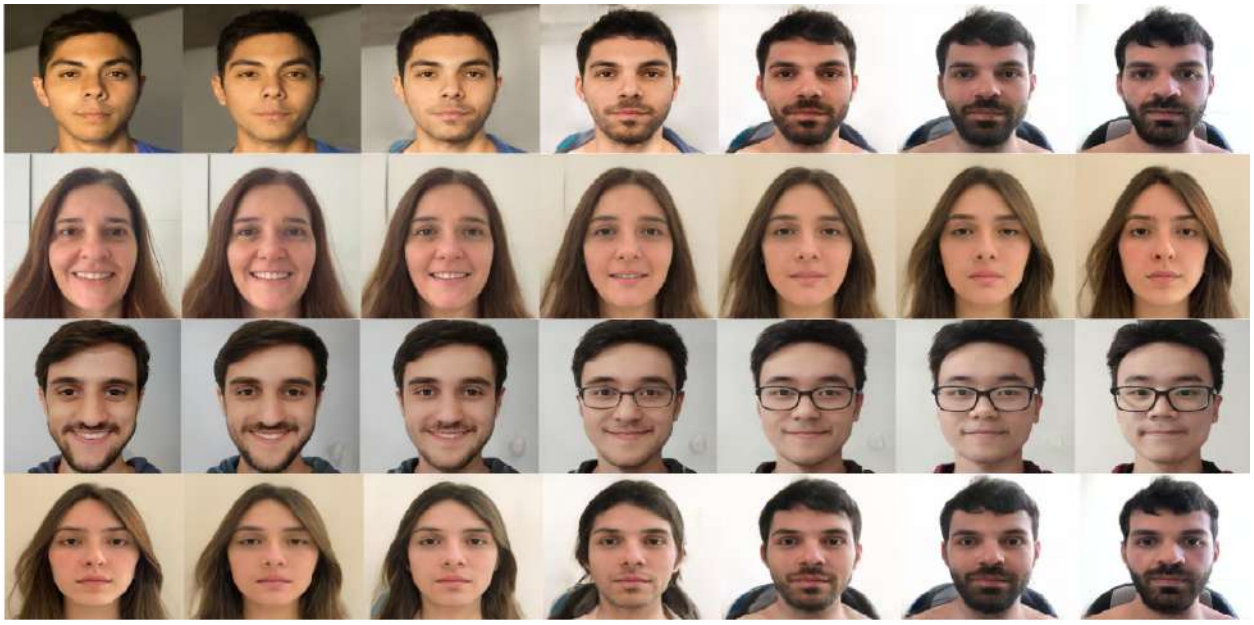
[4] https://github.com/genforce/idinvert

Testing the network, we discovered that it has some difficulties dealing with non “clear” backgrounds, or not aligned faces. Below we can see two examples, the first one the network is able to recreate the background and in the second one it’s not able to recreate the face.

To change some attributes like age, gender and expression, sometimes it works well but sometimes it doesn’t. Analysing the results, we saw that the algorithm doesn't work that well when the person has not so common attributes, like a big beard. Below we can see some results that worked well, the first image is the real person, the second we changed the gender, the second the expression and the third the age.

Below we can see some photos where the gender and the age didn't change well because of the beard, but the algorithm well included eayglasses, because it’s not related to the beard.

To finish the analysis we show the results of the style mixing from real people. The results were quite good. However, we think that if the network had a higher resolution, like 1024x1024 as the styleGAN, the results would be better.



Conclusion

In this project we manipulated in different ways images synthetically generated and also real images. We also analysed some attributes related to those images, such as how attractive a person is, the size of his nose or his lips and others.

We saw that the styleGAN architecture achieved great results to generate synthetical images. More than generating good images, due to the architecture of the network, we are able to manipulate this image in its latent space to create new images related to the original one. We can

do it in two ways. The first way is to take the latent space of another generated image and interpolate both. We saw that this operation leads to great results. The second way is to change a specific attribute, such as the age of the person, or his facial expressions or even the gender. This operation also works well, but it has some flaws when dealing with correlated attributes, such as the use of eyeglasses, which is related to age.

For the second part of this project, we tested the Face Attribute-FAN, a network that is able to predict the attributes (characteristics) of a person. We observed that this network has some bias, especially related to the attractive attribute, which seems to be highly correlated to the color of the person.

As an extra part of the project, we tested the In-Domain Inverse GAN, a network that is able to generate the latent space of real images. We saw that this network gave us good results but it has some flaws when it tries to generate not aligned images or with a "complex" background. Moreover, when changing the person's attributes, we saw that the network has some flaws when dealing with a big beard.

Overall, the results were quite good given the difficulty of the problem. The state of the art for this problem is fastly evolving, as we already have a second version of the styleGAN, which can also perform real image inversions. We learned a lot about the architecture of the networks, how it works and how to use them, but we also know the flaws that we could explore in a future research project in this subject.

References

[1] - He, K., Fu, Y., Zhang, W., Wang, C., Jiang, Y.-G., Huang, F., & Xue, X. (2018). Harnessing Synthesized Abstraction Images to Improve Facial Attribute Recognition. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. https://doi.org/10.24963/ijcai.2018/102

[2] - Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.00453

[3] - Shen, Y., Gu, J., Tang, X., & Zhou, B. (2020). Interpreting the Latent Space of GANs for Semantic Face Editing. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr42600.2020.00926

[4] - Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In Proceedings of European Conference on Computer Vision (ECCV) , 2020