Face Tuning

Group Name: Group 2

Group Members:

First name	Last Name	Student number
Vitor	***Gomes***	***852679***
Alyanna	***Lopez***	***838034***
Mauricio	***Canto Sosa***	***853019***
Kunal	***Badade***	***852958***
Naman	***Shah***	***853267***

Submission date: ***April 22, 2023***

Contents

Abstract

In this project report we design an application pipeline in which a given facial image is manipulated to remove, change, or add characteristics such as glasses, gender, age, etc. Previous work in the topic is minimal and with many of the most relevant solutions being used as proprietary software for big house visual effects companies. We leveraged the power of a PGGAN to generate realistic human faces mapped in the latent space to be able to manipulate them. We performed SVM classification to find different hyperplanes which separate the binary face semantics. With the use of hyperplane normal direction manipulations of the facial noise we were able to successfully manipulate the images and change face attributes with a good precise control between different semantics using conditional manipulation.

Introduction

Facial recognition technology has been advancing rapidly in recent years, and the development of face-tuning technology is no exception. Face-tuning is a type of machine learning technique that involves automatically adjusting and modifying the features of a person's face to make it appear more aesthetically pleasing or socially acceptable. The use of face-tuning technology has become increasingly popular in social media, with many individuals using these tools to enhance their appearance online. This technology uses a range of algorithms to adjust various aspects of a person's appearance, such as the size of their nose, the shape of their lips, or the smoothness of their skin. Face-tuning can be used for a variety of purposes, including enhancing the appearance of selfies, creating digital avatars, re-aging people, and improving the accuracy of facial recognition systems.

The significance of creating face-tuning tool in the entertainment industry is enormous, as it can greatly enhance the storytelling capabilities of filmmakers and provide a more immersive experience for audiences. However, the impact of such a tool is not limited to just this industry. In fact, the potential applications of a face-tuning tool extend far beyond the realm of film and television

One area where it can be beneficial is law enforcement. Facial recognition technology has become a crucial tool for law enforcement agencies in identifying and tracking suspects. However, this technology is limited when it comes to identifying suspects who have aged over time. Face-tuning with the capability of re-aging can help bridge this gap by creating accurate representations of how a suspect may look at a given point in time, even if years have passed since the last known photo (Seiver, 2015) [1]. Another potential application it is in healthcare, particularly in the study of diseases that alter one's facial appearance. By using face-tuning techniques, researchers can study the effects of these diseases on various biological processes and gain a deeper understanding of the mechanisms behind them. This knowledge can then be used to develop new treatments and therapies for such conditions. Additionally, a face-tuning tool can be used in the field of historical preservation. Many historical artifacts, such as paintings, photographs, and sculptures, deteriorate over time. The tool can be used to restore these artifacts to their original condition, preserving them for future generations to enjoy. By using a re-aging tool, historians and preservationists can better understand the historical context of these artifacts and gain new insights into the past.

The development of face-tuning tool is a significant technological advancement that has the potential to improve our understanding of the world around us and enhance our ability to tell stories, both on and off the screen.

Methods

This section presents our steps taken to accomplish face alterations in images for various human characteristics and/or expressions. The general overview of what will be discussed is the approach taken involved training a Progressive Growing General Adversarial Network (PGGAN) on the CelebA dataset to generate images that correlate noise with face semantics in the data. An SVM was then trained for each binary face semantic to learn to predict the binary face semantic given a noise input. This resulted in the creation of a boundary in the noise space that separates the binary face semantics. The latent space was explored with the normal of each SVM hyperplane to alter the noise vector in that direction to tune the faces in each face semantic. The effectiveness of the face tuning neural network was evaluated by comparing the generated images before and after tuning for various face semantics. The following sections provide a detailed description of the data collection and preprocessing steps, the PGGAN and SVM training processes, the latent space exploration approach, and the evaluation methods used in this project.

Dataset and Pre-processing

For the generation of faces with a PGGAN it needed to be trained with a robust dataset of faces which was correctly labeled for the face characteristics and expressions that were aimed to be altered. The dataset chosen for the task was the CelebFaces Attribute Dataset (CelebA) which contains more than 200,000 celebrity images, each with 40 binary attribute annotations. [2] The dataset was chosen above other similar ones due to its large identity diversities, large quantities, and rich annotations.

Due to lack of computing resources and the high computing power needed to work with image data, a substrata of the dataset was extracted to use for the training of the PGGAN. 20,000 images out of the 200,000 was extracted. The face attribute distribution was kept as close to the original dataset.

The last pre-processing step was the normalization of the pixel data of the images. As the images are colored, an RGB normalization was used as seen in Equation 1-3.

$$R' = \frac{R}{R + G + B} \quad (1)$$

$$G' = \frac{G}{R + G + B} \quad (2)$$

$$B' = \frac{B}{R + G + B} \quad (3)$$

With the subset of the dataset created the distribution of the binary attributes was observed to ensure a substantial imbalance was not present. The distribution can be observed in Figure 1.

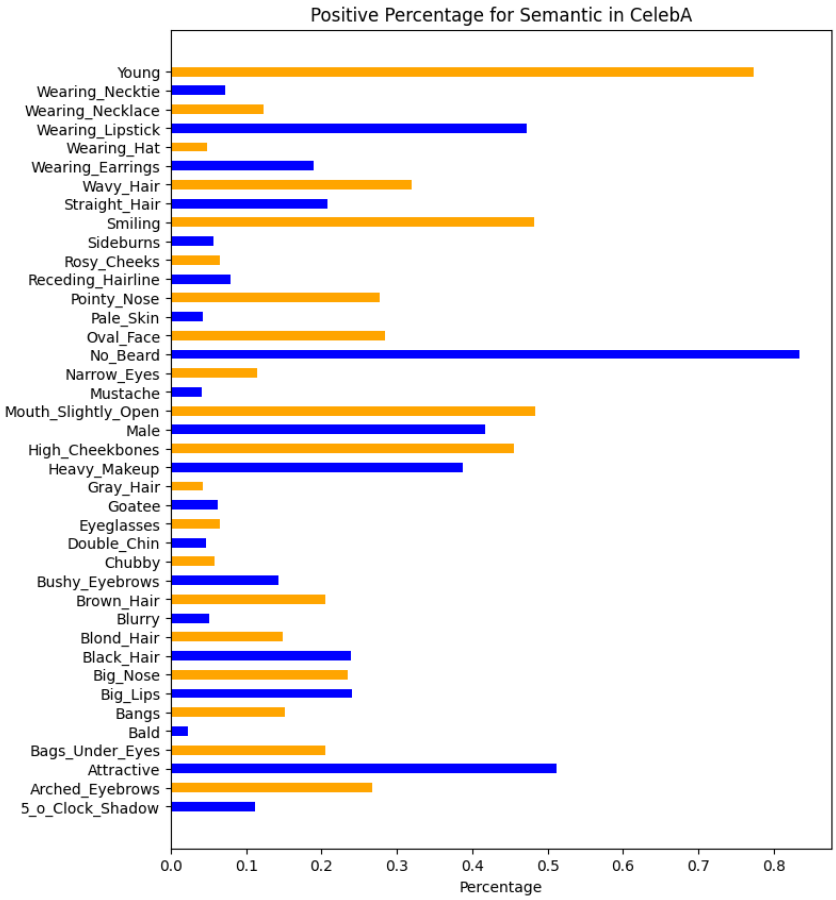


Figure 1 . Distribution of image binary attributes in the CelebA dataset subset used for the PGGAN.[2]

PGGAN Training

The PGGAN was selected over other GAN architectures due to previous works providing promising results in creating realistic face images by training a PGGAN with the CelebA datset.[3]

The training methodology of the PGGAN is progressively increasing the layers through the training for the model to learn and tune the fine details as it trains. This architecture also allows for a faster training time, which is ideal for the limited computing resources and time to complete this project.

To train the PGGAN we followed the training configuration of Gulrajani et al. (2017) [4]. Featuring batch normalization in the generator, layer normalization in the discriminator, and changing the minibatch size of 64 to 32. A minibatch size of 64 would have produced better quality and more realistic face images, but the computing memory for this project was not enough to run a higher minibatch size.

The image generated by the PGGAN were relatively natural, with some specific instance in which some blurring and meshing could be observed. The results images are satisfactory and have enough sharpness and distinguishable attributes to be used to train the SVM. We had previously tried to train the PGGAN with a minibatch size of 16 which resulted in unnatural images.

This resulting PGGAN was used to generate 100,000 face images to train the SVM's. We additionally trained some classifiers with the CelebA datset to classify the binary attributes of the PGGAN's generated images. The distribution of said images can be observed and compared to the original CelebA dataset attributes in Figure 2.

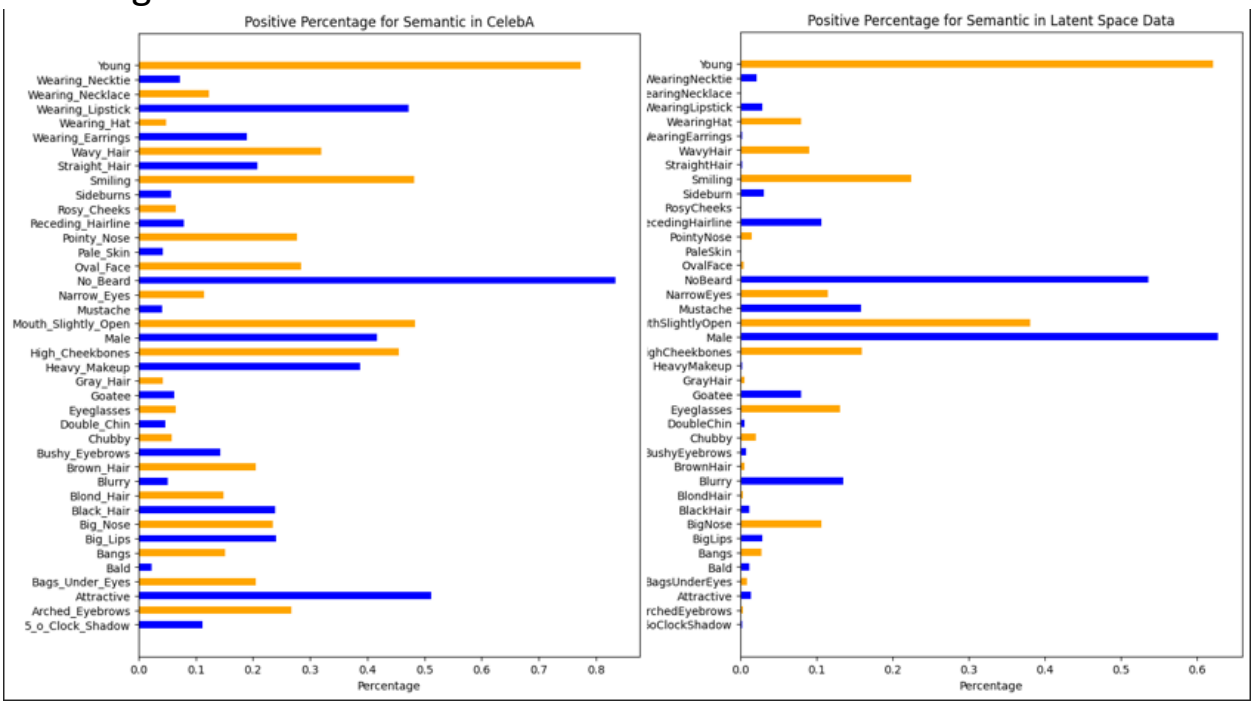


Figure 2 . Distribution of image binary attributes in the CelebA dataset subset (left) and the distribution of attributes in the PGGAN generated images(right).

Face Semantics SVM's Training

The PGGAN generated images were created with the purpose of training various SVM's to find the hyperplane that separates the binary attributes in the latent space. These binary attributes in the latent space can be denoted as semantic information.

The linear interpolation between semantic scores of different noise (face images) forms an specific direction that further defines the hyperplane; therefore, for any binary semantic, there exists a hyperplane in the latent space which creates a boundary of said semantic. [5]

With the training of different SVM's it was possible to find the hyperplanes which create the boundaries of the binary semantics. The SVM's were created to find the hyperplanes for all of the binary attributes listed on the CelebA dataset and explore the latent space on that basis.

Latent Space Exploration and Manipulation

In this part, it is explained how to use the semantics in the latent space and the SVM's for image manipulation.

With the found hyperplanes located in the latent space there is a boundary between a binary semantic (e.g male - female). If this hyperplane creates a separation between the binary semantic, that would also mean that moving one noise point from one side to the other should change it's semantic attribute separated by the hyperplane. To find this direction to move on the normal of the hyperplane was calculated which would give us a perpendicular direction along the hyperplane to manipulate the semantic it's representing.

As more than one semantic is present, the manipulation of the image via the hyperplane normal may affect other semantics present as some of them can be coupled between each other. To achieve a more precise control conditional manipulation [5] was used. With conditional manipulation two hyperplane normal can be used to project a direction from one to the other to be able to manipulate one semantic without affecting the other as seen in Figure 3.

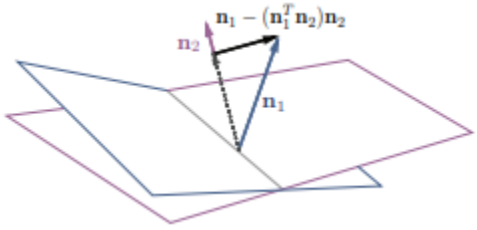


Figure 3 . Illustration of the conditional manipulation in subspace. The projection of n1 onto n2 is subtracted from n1, resulting in a new direction. [5]

Results

Qualitative and quantitative statistics were used to evaluate our face-tuning model. Figure 4 shows the unmodified images the GAN generated.

Figure 4 . original images generated by the GAN

Qualitative Evaluation

As mentioned in the methodology, face-tuning was done by performing exploration on the different features in the latent space distinctly separated by the SVM classifier. Here, exploration on age, gender, and facial features were performed.

Figure 5 (where the single image represents the unmodified face), shows the output images when the model traversed the age direction and discarding the projection along the gender direction. in the latent space. As observed, the model was able to transform the original image into looking older and younger (from left to right, top to bottom). The major changes distinctly for male when aged older were facial hair growth and hair loss. Meanwhile, both genders experienced loss of muscle tone and thinning skin giving off a drooping appearance.

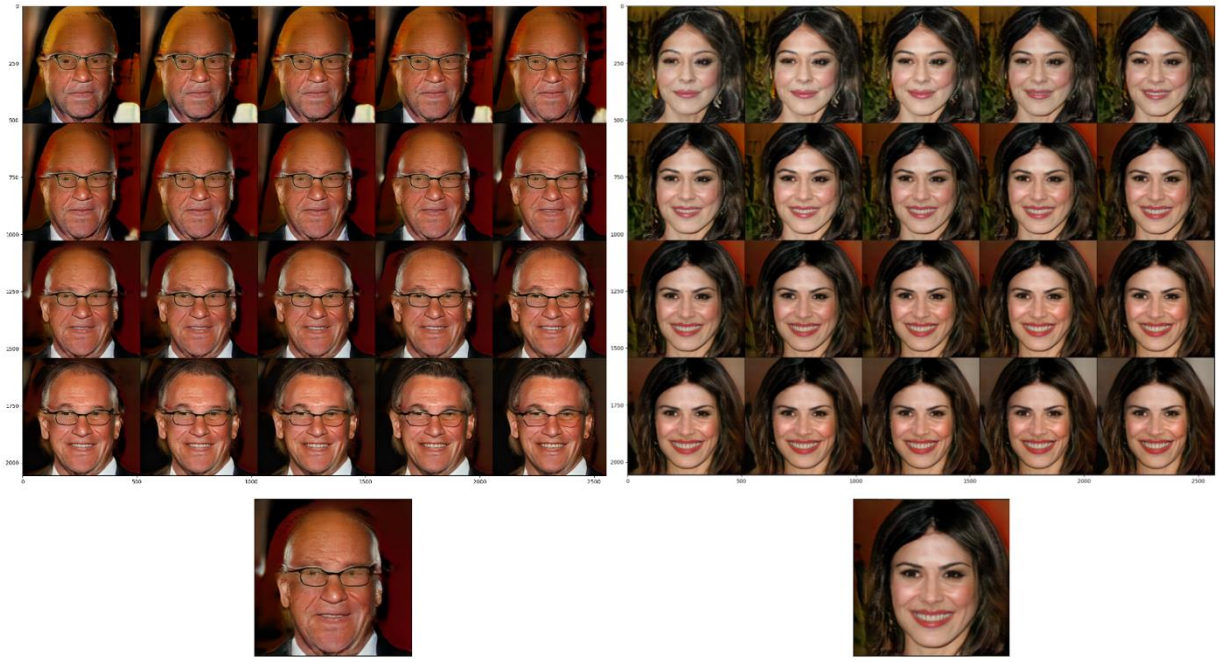


Figure 5. Age Feature Exploration

Additional experiments were made by removing more projections on different features. For example, Appendix A displays the transformation of a male along the age direction but overlaps on other feature directions like gender, baldness, and presence of lipstick. By doing this, the model generated more refined results. The age-related facial changes mentioned earlier still show on the images. In addition, referring to Figure 6 and using the same conditions, the model removed the man's eyeglasses as he aged younger.

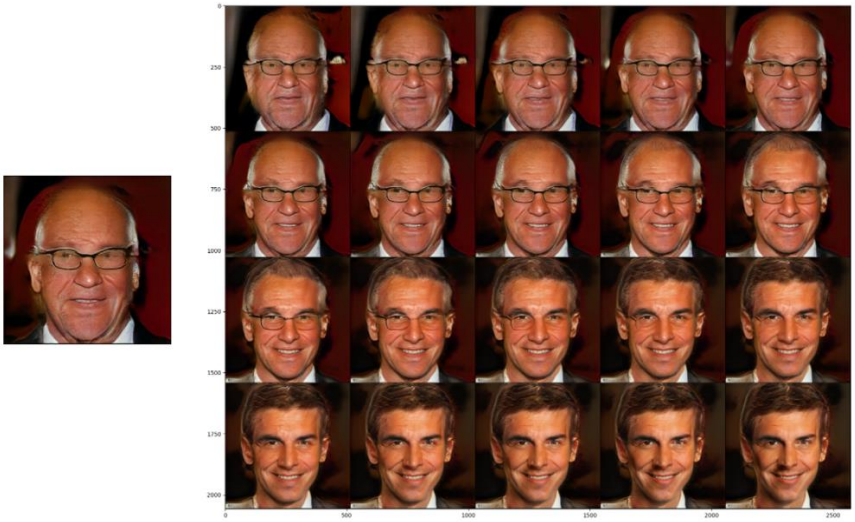


Figure 6. Multiple feature exploration.

Quantitative Evaluation

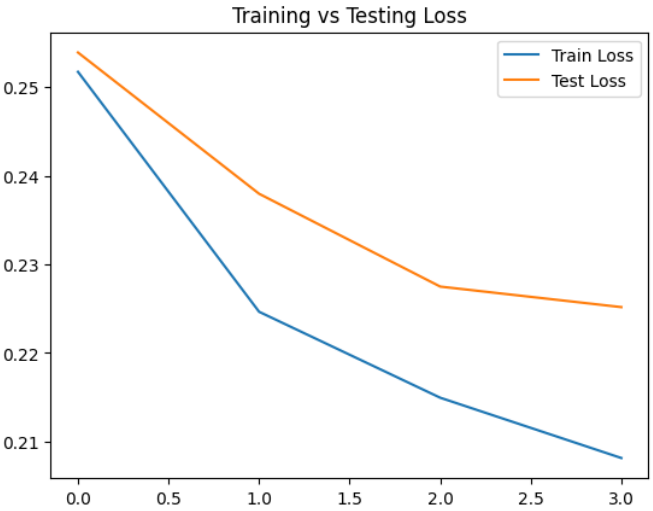
Evaluation metrics used for this research are accuracy, f1-score, and MSE loss. The accuracy score is a metric used to evaluate the performance of a classification model. It measures the proportion of correct predictions among all the predictions made by the model (Google Developers, 2022). [6] Another classification metric commonly used is f1-score, which combines two other evaluation metrics: *precision* and *recall* into a single metric that balances both measures. Precision refers to the percentage of correct positive predictions the model made while Recall tells how good the model can predict which ones are true positives (Korstanje, 2021). [7] MSE loss measures the average of the squared differences between the predicted and actual values of a model's output. This metric is widely used in regression tasks and is useful for evaluating the performance of a model's predictions (IBM, 2023). [8] It is a non-negative value that indicates how well a model can fit the data. The lower the value of MSE, the better the model's performance.

Figure 7 shows the SVM classifier evaluation results. Looking at the training and testing accuracies, the classifier did not experience overfitting nor underfitting. The model did, however, performed badly on most features based on the f1-scores calculated. The features that had higher f1-scores (>= 0.5) than the rest are baldness, gender, age, and presence of facial hair, which were some of the variables that were explored during testing as mentioned in Qualitative Evaluation.

Feature	Data Size	Training Accuracy	Testing Accuracy	F1 Score	Feature	Data Size	Training Accuracy	Testing Accuracy	F1 Score
5oClockShadow	2630.0	1.000000	0.815589	0.291971	Male	100000.0	0.606062	0.597050	0.668340
ArchedEyebrows	3160.0	0.919304	0.825949	0.246575	MouthSlightlyOpen	100000.0	0.605088	0.595150	0.494916
Attractive	13620.0	0.834159	0.813877	0.274678	Mustache	100000.0	0.756850	0.756900	0.299020
BagsUnderEyes	8580.0	0.856498	0.812937	0.230216	NarrowEyes	100000.0	0.811987	0.809150	0.290652
Bald	11800.0	0.899894	0.883898	0.534014	NoBeard	100000.0	0.589013	0.587000	0.611074
Bangs	27980.0	0.870756	0.865618	0.448680	OvalFace	4000.0	0.945625	0.843750	0.418605
BigLips	29140.0	0.840125	0.832361	0.168511	PaleSkin	460.0	1.000000	0.663043	0.114286
BigNose	100000.0	0.815050	0.810900	0.193947	PointyNose	15330.0	0.842792	0.805610	0.227979
BlackHair	11640.0	0.868342	0.843213	0.396694	RecedingHairline	100000.0	0.831013	0.827150	0.346873
BlondHair	3440.0	0.967297	0.824128	0.324022	RosyCheeks	900.0	1.000000	0.800000	0.379310
Blurry	100000.0	0.809737	0.808150	0.333738	Sideburn	30490.0	0.848434	0.846343	0.322487
BrownHair	5430.0	0.894337	0.829650	0.381271	Smiling	100000.0	0.758250	0.750750	0.490963
BushyEyebrows	7270.0	0.883425	0.834250	0.343324	StraightHair	2360.0	1.000000	0.817797	0.245614
Chubby	20230.0	0.858440	0.842808	0.354970	WavyHair	90420.0	0.820090	0.813094	0.220839
DoubleChin	5070.0	0.895710	0.816568	0.311111	WearingEarrings	2580.0	0.986919	0.773256	0.214765
Eyeglasses	100000.0	0.816388	0.816800	0.351734	WearingHat	79550.0	0.840383	0.839786	0.269837
Goatee	80290.0	0.830100	0.824387	0.197952	WearingLipstick	28540.0	0.848108	0.843027	0.208481
GrayHair	5280.0	0.888021	0.817235	0.322807	WearingNecklace	610.0	1.000000	0.729508	0.232558
HeavyMakeup	2320.0	0.994612	0.765086	0.167939	WearingNecktie	21500.0	0.860465	0.838837	0.371714
HighCheekbones	100000.0	0.799750	0.794300	0.445104	Young	100000.0	0.644550	0.632850	0.683532

Figure 7. SVM Classifier evaluation result

Image classifiers were evaluated using accuracy scores and MSE loss. Similar to the SVM classifier, no overfitting nor underfitting happened. The image classifiers yielded good accuracies ranging from 0.8 to 0.97 wherein gender had the highest value of 0.97, as shown in Figure 8. Moreover, an MSE loss of around 0.22 was obtained.



Label	5oClockShadow	ArchedEyebrows	Attractive	BagsUnderEyes	Bald	Bangs	BigLips	BigNose	BlackHair	BlondHair
TrainingAcc	0.8947	0.8765	0.9597	0.8763	0.9648	0.9398	0.9532	0.9568	0.9721	0.9169
TestAcc	0.8568	0.8539	0.9374	0.8568	0.9583	0.8539	0.9380	0.9305	0.9623	0.9065
Label	Blurry	BrownHair	BushyEyebrows	Chubby	DoubleChin	Eyeglasses	Goatee	GrayHair	HeavyMakeup	HighCheekbones
TrainingAcc	0.9490	0.8767	0.8427	0.9550	0.9443	0.8981	0.8881	0.9077	0.8644	0.8558
TestAcc	0.9346	0.8053	0.8026	0.9244	0.9267	0.8694	0.8402	0.8447	0.8494	0.8089
Label	Male	MouthSlightlyOpen	Mustache	NarrowEyes	NoBeard	OvalFace	PaleSkin	PointyNose	ecedingHairlin	RosyCheeks
TrainingAcc	0.9811	0.8950	0.8726	0.8681	0.9459	0.8511	0.8336	0.8811	0.9686	0.9260
TestAcc	0.9732	0.8646	0.8251	0.8274	0.9379	0.8414	0.8030	0.7220	0.9062	0.8600
Label	Sideburn	Smiling	StraightHair	WavyHair	aringEarrin	WearingHat	WearingLipstick	WearingNecklace	WearingNecktie	Young
TrainingAcc	0.9191	0.9138	0.8728	0.9400	0.8712	0.9678	0.8765	0.9101	0.9493	0.9127
TestAcc	0.8739	0.9064	0.8531	0.8614	0.8495	0.9198	0.8458	0.9069	0.8800	0.8947

Figure 8. Image classifier evaluation results: MSE Loss (top) and Accuracy (bottom)

Conclusions and Future Work

Using the created faces by the PGGAN we were able to interpret the semantics in the latent space of GANs. By manipulating those semantics using the hyperplanes that separate them the proposed face manipulation was attained changing facial attributes of GAN generated facial images.

The manipulation only works with GAN generated faces, as these already have a space in the latent space. To perform any kind of manipulation of a real image in the latent space we would need to explore more techniques to reconstruct the images with minimal loss, or to use an encoder to invert the image into the latent space.

References

[1] Seiver, S. (2015) What would you look like 20 years from now?, The Marshall Project. The Marshall Project. Available at: https://www.themarshallproject.org/2015/06/25/what-would-you-look-like-20-years-from-now (Accessed: April 21, 2023).
[2] Z. Liu, P. Luo, X. Wang, and X. Tang, 'Deep Learning Face Attributes in the Wild', CoRR, vol. abs/1411.7766, 2014.
[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, 'Progressive Growing of GANs for Improved Quality, Stability, and Variation', CoRR, vol. abs/1710.10196, 2017.
[4] Ishaan Gulrajani, Faruk Ahmed, Mart´ın Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs. CoRR, abs/1704.00028, 2017.
[5] Y. Shen, J. Gu, X. Tang, and B. Zhou, 'Interpreting the Latent Space of GANs for Semantic Face Editing', CoRR, vol. abs/1907.10786, 2019.
[6] Google, "Classification: Accuracy | Machine Learning Crash Course," *Google Developers*, 2019. https://developers.google.com/machine-learning/crash-course/classification/accuracy
[7] J. Korstanje, "The F1 score," *Medium*, Aug. 31, 2021. https://towardsdatascience.com/the-f1-score-bec2bbc38aa6
[8] "Mean squared error," *www.ibm.com*, Apr. 16, 2021. https://www.ibm.com/docs/en/cloud-paks/cp-data/3.5.0?topic=overview-mean-squared-error (accessed Apr. 22, 2023).
[9] G. Zoss, P. Chandran, E. Sifakis, M. Gross, P. Gotardo, and D. Bradley, 'Production-Ready Face Re-Aging for Visual Effects', ACM Transactions on Graphics, vol. 41, pp. 1–12, 11 2022.