Facial Emotion Editing and Transfer with Pretrained Networks

Ali Azak, Murat Sahin

***Abstract*—Generative Adversarial Networks (GANs) have emerged as a powerful tool for image generation and editing, especially with the development of StyleGAN. In this work, we utilized different pretrained networks to edit emotions of a given face image and also transfer emotions between two given face images. We utilized the StyleGAN model as our generator. Two different architectures were used to encode images into the StyleGAN latent space. We employed an InterfaceGAN-type methodology to identify emotional directions in the StyleGAN latent space, with the help of an emotion detection network, to facilitate emotional editing. Additionally, we developed a successful method that combines GAN inversion based normalization and vector arithmetic for transferring emotion from a given image to another.**

I. Introduction

GENERATIVE Adversarial Networks (GANs) have revolutionized the field of image generation and manipulation, offering incredible capabilities in producing high-quality and diverse visual content. Among the various GAN architectures, StyleGAN has emerged as a particularly powerful model, enabling fine-grained control over generated images through its innovative style-based design. In this project, we leveraged the capabilities of StyleGAN to explore two key tasks: facial emotion editing and emotion transfer between images.

In our work, we have discretized human emotions into eight categories: anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. We focus on utilizing pretrained networks to edit the emotional expressions of given facial images and to transfer emotions from one face to another. We utilize StyleGAN's latent space to encode and manipulate facial images, employing GAN inversion techniques to accurately map real images into this space. By identifying emotional directions within the latent space through an InterfaceGAN-type approach, we can perform emotional edits on facial images. Furthermore, we develop a method that combines GAN inversion based normalization with vector arithmetic to achieve effective emotion transfer between facial images.

The rest of the paper is organized as follows: In Section II, we explain the related work regarding the models we have used and the tasks we aimed to accomplish. In Section III, we detail our experimental process, and in Section IV, we discuss the results.

II. Related work

A. GAN Architectures

GANs are improving and advancing since the first GAN study [2].

1) DCGAN: Using GANs with convolutional networks started with DCGAN [11]. This model consists of two CNN networks trained adversarially to capture the data distribution and estimate the probability that a sample came from the training data rather than the generator. The DCGAN model also examined latent space manipulation and interpolation, revealing that the latent space of GAN architectures are quite useful for making edits.

2) ProgressiveGAN: ProgressiveGAN [5] trains the generator and discriminator in a progressive manner. Starting with lower resolutions, the model gradually increases the resolution, enabling it to create fine and coarse details in the generation process.

3) StyleGAN: StyleGAN [8] is based on the ProgressiveGAN architecture. The name is inspired by style transfer networks. In traditional GAN networks, a random latent z is used to start the generation process, introducing randomness. In the StyleGAN architecture, the random latent z is first mapped into w space. This 'style,' along with noise, is injected into every resolution in the generation process. The authors explored the effect of different style injections at various resolutions and discovered that novel images could be generated using different styles w for fine and coarse resolutions. In the subsequent paper, StyleGANv2 [9], the authors removed the Adaptive Instance Normalization modules used in the style injection process and instead employed a new weight modulation technique, which they claim is superior.

B. GAN Inversion

GAN inversion is a technique to map real images into the GAN latent space, specifically StyleGAN, to modify this latent space for image manipulation and editing. There are two primary methods for GAN inversion: encoder-based and optimization-based methods. Encoder-based methods are fast but not generalizable to unseen samples. Optimization-based methods, while slower, can adapt to new samples.

1) pixel2style2pixel: The pSp [12] paper introduces an encoder-based GAN inversion method. The architecture uses a standard feature pyramid over a ResNet backbone. For each feature map in the pyramid in the extended latent space W^+, an encoder 'map2style' network, a small CNN, is employed. The resulting style is used in the same manner as in the StyleGAN architecture.

2) High Fidelity GAN Inversion: High-Fidelity GAN Inversion [15] is an optimization-based GAN inversion method. Given that low-rate latent codes have low fidelity but high editability and high-rate latent maps have the opposite, the authors propose a network to combine both latent codes for high fidelity and editability. They fuse the latent codes and maps in the generation process and optimize the encoder network without altering the generator parameters.

3) HyperStyle: HyperStyle [1] is a hybrid of optimization-based and encoding-based inversion methods, based on the pSp architecture. On top of the pSp architecture, HyperStyle modifies the pretrained generator's weights to optimize the generator for better inversion. It achieves the optimization by adding offsets to each channel of the StyleGAN architecture rather than modifying each parameter separately, significantly reducing the process complexity.

C. Editing with StyleGAN

Editing with StyleGAN is generally performed with vector operations in the StyleGAN latent space. Supervised edit methods use SVM-like classifiers to find meaningful directions in the StyleGAN latent space. Once these directions are identified, vector operations along these directions can be performed to achieve desired edits. InterfaceGAN [14] works this way. Unsupervised edit methods use PCA-like methods to find meaningful directions. GANSpace [4] performs dimension reduction to find the strongest eigenvectors, which can then be used to explore the latent space and observe their effects on images.

The TensorGAN architecture [3] performs emotion editing by factorizing the latent space $W+$ of the StyleGAN architecture into meaningful subspaces. After inverting the images into latent space, tensor manipulation is used to edit the emotions of the faces. Assuming D dimensioned latent codes for P different persons, performing E expressions each with I different intensities from R different rotations, the data is arranged into a 5th order tensor $T^{DxPxExRxI}$. Higher-Order Singular Value Decomposition is then performed to derive the corresponding directions and magnitudes for linear interpolation of desired attributes.

(a) Sample from FFHQ dataset

(b) Sample from CelebA dataset

Fig. 1: Samples from both datasets

D. Style Transfer with StyleGAN

DualStyleGAN [16] accomplishes the portrait style transfer task using two encoders to encode both input and style images into the StyleGAN latent space, injecting style into the StyleGAN architecture in the conventional way. The color and structure are controlled by injecting different styles at fine and coarse resolutions.

III. Experiments

A. Datasets

The pretrained models utilized in this project were all trained on the FFHQ [7] dataset. For our experiments involving GAN inversion, images were selected from the CelebA [6] dataset to prevent the introduction of information from the training set. Figure 1 illustrates the samples from both datasets.

B. Methodology

This section will present the methodology employed during the project's development. The emotion editing process will serve as the foundation for the subsequent emotion transfer. Figure 2 provides an overview of the overall architecture. We will provide a detailed description of each module.

1) GAN Inversion: The methodology begins with the finding of the latent code of a given image within the StyleGAN latent space. This is achieved through the utilisation of GAN inversion techniques, namely pSp and HyperStyle. While the former is capable of inverting images with satisfactory results and the retention of shallow identity, the latter is more effective in terms of inversion, with the addition of additional corrections to the code during the generation process. Figure 3 illustrates the image generations obtained from both methods, demonstrating that HyperStyle is more successful. Both methods are utilized in this work.

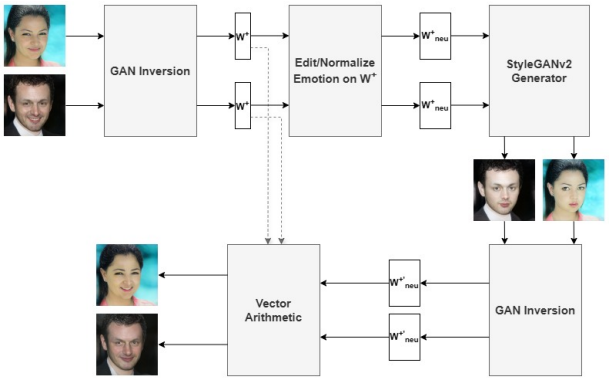


Fig. 2: Designed architecture

(a) Original (b) pSp (c) HyperStyle

Fig. 3: Inversion quality comparison

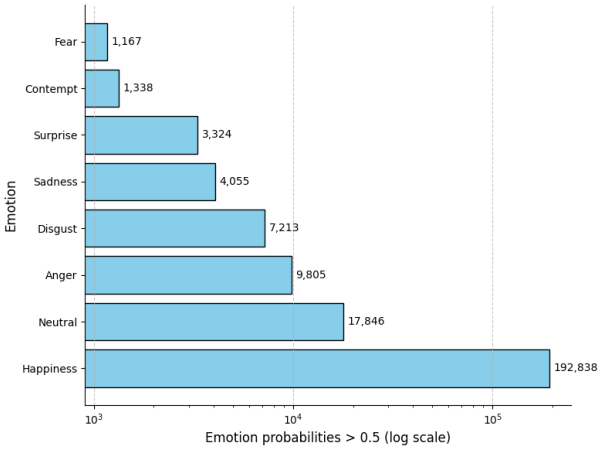


Fig. 4: Probabilities exceeding threshold for each emotion

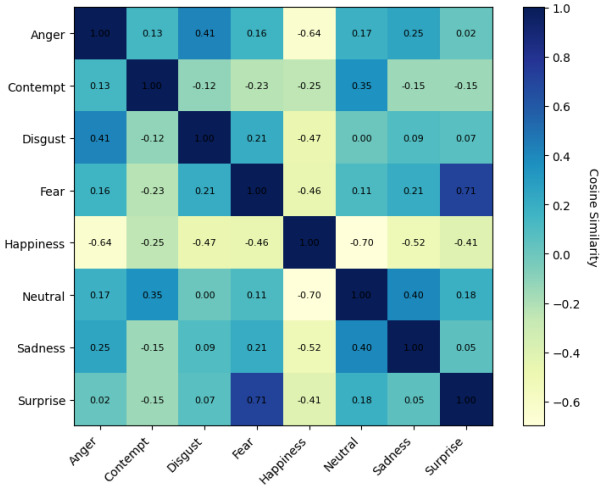


Fig. 5: Similarity between emotions

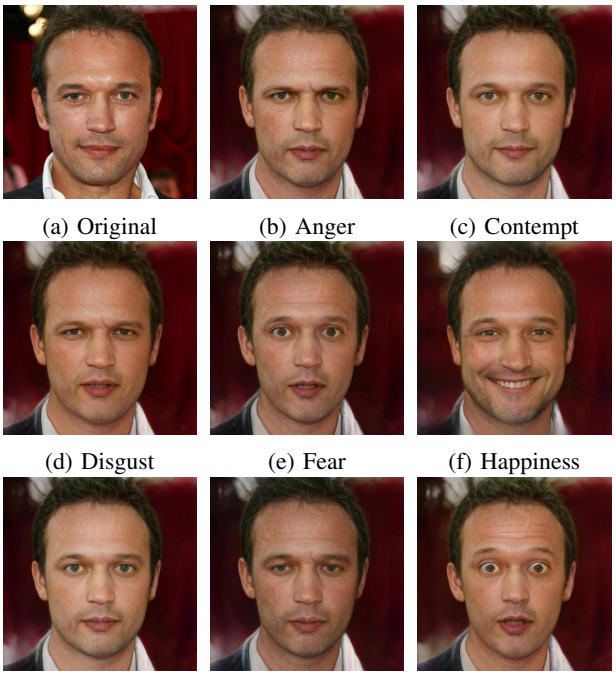
2) Finding Emotional Directions in Latent Space: In order to identify the emotion directions in the latent space, we employed a methodology similar to that employed by InterfaceGAN. Initially, we generated 300,000 images, after which we utilized a pretrained emotion detection network [13] to calculate emotion scores for each of the eight emotions: anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. Subsequently, the authors of InterfaceGAN proposed selecting 10,000 extreme examples for each class and training an SVM classifier on them. However, we had to choose a different approach.

The proposed approach was not applicable in our context. Due to the nature of FFHQ dataset, generated images are not representing the emotion space as a whole. Please refer to Figure 4 for a visual representation of the dominant emotions in generated images. Happiness is the dominant emotion for the majority of images, while emotions such as fear are significantly underrepresented. Due to this, we trained eight distinct SVM regressors for each emotion, utilizing the entire data set with labels as softmax outputs for that specific emotion from the emotion detector. This approach proved effective in capturing the desired emotions. Coefficients of the regressor are extracted to obtain directions.

Additionally, the authors proposed the removal of correlations between directions, yet in our case, we found it beneficial to retain these correlations. Figure 5 illustrates the degree of similarity between different emotional directions. The correlation between different pairs of emotions, such as fear and surprise, is notable. This is despite the fact that they are detected via similar expressions, which would be expected to result in a high degree of correlation.

3) Emotional Editing in Latent Space: Once the directions have been obtained, the process of editing emotions is relatively straightforward, as illustrated by Equation 1 where C denotes latent code (W^+ space). Figure 6 represents the resulting edits in response to the obtained emotion directions. As a further point of interest, we have verified that the neutral direction is identical to the sum of all other directions. This demonstrates that our method is both logical and effective.

$$\mathrm{C}_{\mathrm{edit}} = \mathrm{C} + \mathrm{k} \times \mathrm{Direction} \tag{1}$$



(a) Original (b) Anger (c) Contempt

(d) Disgust (e) Fear (f) Happiness

(g) Neutral (h) Sadness (i) Surprise

Fig. 6: Emotion editing results, inversion by pSp

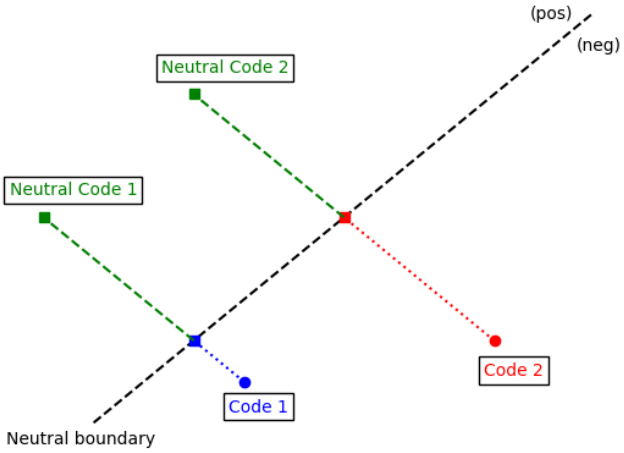


Fig. 7: Visualization of the neutralization in 2D

Fig. 8: Visualizing the transfer arithmetic

4) Emotion Transfer: In order to achieve the transfer of emotion, there is a preliminary step in our architecture. We are using the neutralized version of subjects' faces to learn the emotional representation of the subject with the target emotion. Normalizing process is relatively straightforward. For a given code, we first project it to the neutral boundary of SVM and then add the neutral dimension multiplied by a constant k (k=20 for our experiments). This can be seen in Equation 2, where C denotes latent code, ND denotes neutral direction and f is the projection function. Also 2D visualization of this process can be observed in Figure 7. The initial projection is necessary to prevent images from being over- or under-neutralized, particularly in the context of emotion transfer, where the process involves pairs. Therefore, it is important to ensure consistency.

$$\mathrm{C}_{\mathrm{neu}} = f(\mathrm{C}, \mathrm{ND}) + \mathrm{k} \times \mathrm{ND} \tag{2}$$

Once neutral codes have been obtained, it is reasonable to assume that vector arithmetic can be applied to facilitate emotion transfer. However, this approach does not yield the desired outcome. Direct subtraction of the neutral code of the target emotion from its original and subsequent addition to the neutral source identity code results in a nearly identical outcome to that of the initial source identity image. Our hypothesis is that the normalization process may have inadvertently altered the nature of the latent code, and although the expected neutralization is visually achieved, the addition of the neutral direction suppresses any other information, rendering the code unusable for editing or transfer purposes. We identified a solution to this issue. Initially, we attempted to normalize the latent code with various techniques, but this proved unsuccessful. However, generating images from the normalized code and inverting them back worked well. This provided us with a code that could generate nearly identical images with enhanced meaning for use. Performing the exact same vector arithmetic yielded the desired output. Please refer to Figure 10 for the resulting images with and without the inversion process. This inversion acts as a normalizer, preventing the neutral direction from suppressing other emotions. The transfer code calculation is defined by Equation 3, where TC represents the target emotion code, TC'_{neu} denotes the inverted neutral target emotion code, and SC'_{neu} stands for the inverted neutral source identity code. You may also see the visualization of this equation in Figure 8. Additionally, we attempted to train a network for normalization, but were unable to achieve a performance level comparable to inversion.

$$\text{Transfer Code} = (TC - TC'_{\mathrm{neu}}) + SC'_{\mathrm{neu}} \tag{3}$$

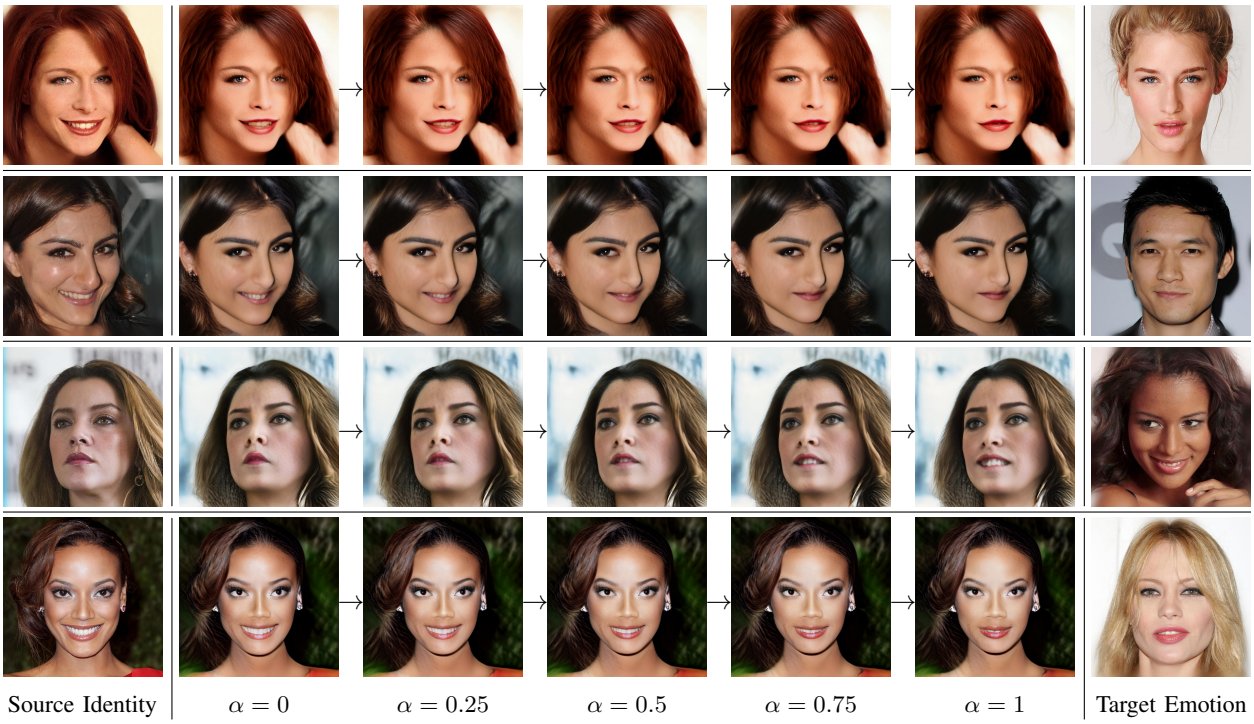


Fig. 9: Emotion transfer results with interpolations, inversions by HyperStyle

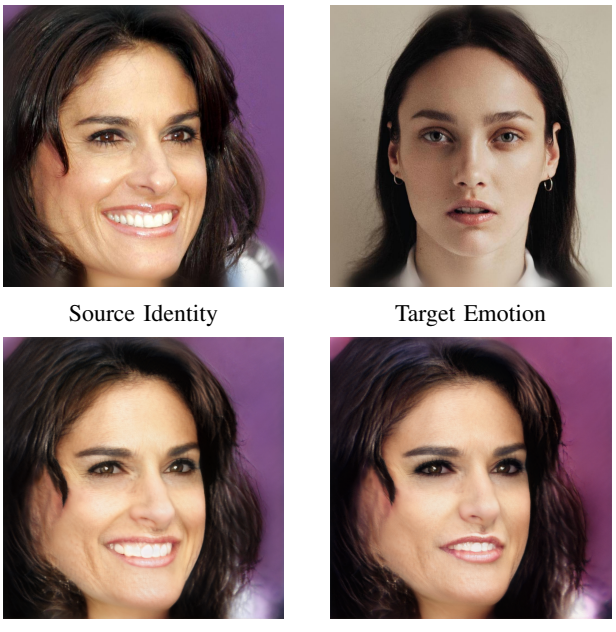


Fig. 10: Emotion transfer result with and without inversion

IV. Results

A. Evaluation

For evaluating our methodology, we used FID score and we defined an emotion score (ES). Emotion score is defined as the average cosine similarity between the emotion prediction of the target emotion image and emotion predictions of the multiple source identity images after the transfer.

We first took an arbitrary image, and edited its emotion with each of our directions, can also be seen in forementioned Figure 6. Then, we sampled 2000 images from the CelebA dataset and for each sample, we ran our emotion transfer pipeline with each target emotion image and saved the generated images. Then, we calculated FID scores using [10] and emotion scores for each emotion seperately. You can see the obtained results in Table I. It can be seen that HyperStyle is more successful at inverting images, since FID scores are generally lower. However, there is not a clear winner in terms of editability with our method. Inversion with pSp had a great emotion score for emotions: Anger, Contempt, Disgust, Happiness (with a great margin), however, inverting with HyperStyle proved to be a more effective approach for the remaining emotions. It should be noted that this score may not fully reflect human perception, as it is only based on a pretrained emotion detection network.

	HyperStyle		pSp	
Emotion	ES ↑	FID ↓	ES ↑	FID ↓
Anger	0.371	36.385	0.401	37.590
Contempt	0.707	36.926	0.712	39.999
Disgust	0.539	35.868	0.565	39.156
Fear	0.541	36.383	0.410	35.243
Happiness	0.733	34.348	0.961	41.131
Neutral	0.770	36.214	0.587	39.738
Sadness	0.778	37.336	0.644	34.875
Surprise	0.123	35.202	0.086	34.752
Original		22.558		26.045

TABLE I: FID Scores and Emotion Scores (ES) Using HyperStyle and pSp Inversion Methods for Different Emotions

Furthermore, for further visualizing the transfer results, we sampled some example source and target image pairs, and this time, we did the transfer while interpolating the emotional differences. This was achieved through the application of the formula presented in Equation 4, with the resulting visualization presented in Figure 9.

$$\begin{aligned} \text{Transfer Code} &= \alpha \times (TC - TC'_{\text{neu}}) \\ &+(1-\alpha) \times (SC - SC'_{\text{neu}}) + SC'_{\text{neu}} \end{aligned} \tag{4}$$

B. Discussion

Generally, style injection based methods would be more suitable for this type of tasks, however we were successfully able to transfer emotions between two given images with an editing approach. However, there are some drawbacks. Inversion based normalization method is inevitably losing some information and adding additional complexity. Finding a way to numerically normalize the latent codes might be better.

V. Conclusion

In this project, we successfully achieved our objectives of finding emotional directions in the StyleGAN latent space, performing popular style editing, and developing a great method for emotion transfer — a relatively unexplored research area. By employing various inversion techniques with StyleGAN2, and utilizing only pretrained networks, we were able to identify and manipulate emotional expressions in human faces effectively. The results of our experiments demonstrate the potential and versatility of GANs in the domain of emotion editing and transfer.

References

[1] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] René Haas, Stella Graßhof, and Sami S Brandt. Tensor-based emotion editing in the stylegan latent space. *arXiv preprint arXiv:2205.06102*, 2022.

[4] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020.

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[6] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2017.

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[9] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[10] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11400–11410, 2022.

[11] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[12] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[13] A. Savchenko, Lyudmila V. Savchenko, and Ilya Makarov. Classifying emotions and engagement in online learning based on a single facial expression recognition neural network. *IEEE Transactions on Affective Computing*, 13:2132–2143, 2022.

[14] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020.

[15] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022.

[16] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *CVPR*, 2022.