FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Pose and Expression transfer between face images
Student:	Petr Jahoda
Supervisor:	Ing. Jan Čech, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

The objective of this thesis is to develop a method for transferring the pose and expression of a source (driving) face image to a target (identity) face image, while preserving the identity of the target face [1,2]. The proposed method should be able to handle changes in pose and expression in close to real time, and should be able to generate realistic images.

It is recommended to use pre-trained GAN models that produce high-quality photo-realistic output, e.g., [3,4], in a similar spirit as in [5], where a generative part of the network is fixed and only an encoder is trained. As a side effect, the method should serve as another generative model that would allow to synthesise random identities with independently controllable pose and expression.

(1) Make a literature survey.
(2) Design a neural network architecture and a training procedure that learns from a suitable dataset, e.g., VoxCeleb2 [6].
(3) Evaluate the method on an independent set, e.g., a test split of VoxCeleb2. Measure both pose/expression transfer fidelity and face identity preservation.

References

---

1. Drobyshev, Nikita and Chelishev, Jenya and Khakhulin, Taras and Ivakhnenko, Aleksei and Lempitsky, Victor and Zakharov, Ego. MegaPortraits: One-shot Megapixel Neural Head

---

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 26 January 2023 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Avatars. In Proc. ACM International Conference on Multimedia, 2022.
2. Ting-Chun Wang, Arun Mallya, Ming-Yu Liu. One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. CVPR, 2021.
3. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. Proc. In Proc. CVPR, 2020.
4. Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient Geometry-aware 3D Generative Adversarial Networks. In Proc. CVPR, 2022.
5. Adela Subrtova, Jan Cech, Vojtech Franc. Hairstyle Transfer between Face Images. In Proc. IEEE Conference on Automatic Face and Gesture Recognition, 2021.
6. J. S. Chung, A. Nagrani, A. Zisserman. VoxCeleb2: Deep Speaker Recognition. In Proc. INTERSPEECH, 2018.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Pose and Expression transfer between face images

Petr Jahoda

Department of Applied Mathematics
Supervisor: Ing. Jan Čech, Ph.D.

May 10, 2023

Acknowledgements

I would like to express my sincere gratitude to my supervisor Ing. Jan Čech, Ph.D. for his patient guidance and willingness to devote his time to this work. I would like to extend my sincere appreciation to the Center for Machine Perception at FEE CTU for generously providing me with access to their servers for storing and training the models.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on May 10, 2023

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Jahoda, Petr. *Pose and Expression transfer between face images.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Abstract

This thesis proposes a method for pose and expression transfer between face images. Given a source and target face portrait, the designed network produces an output image where the pose and expression from the source face image are transferred onto the target identity. The architecture consists of two encoders and a mapping network that maps the two inputs into the latent space of StyleGAN2, which generates a high-quality image. The training is self-supervised without the need for labeled data. Our method achieves close to real-time image generation while also enabling the synthesis of random identities with independently controllable pose and expression.

Keywords human face generation, expression transfer, pose transfer, StyleGAN2, deep learning

Abstrakt

Tato práce představuje metodu pro přenos pózy a výrazu mezi portréty. Po zadání dvou obrázků obličeje, zdrojového a cílového, navržená síť vygeneruje portrét, kde jsou póza a výraz z obrazu zdrojového obličeje přeneseny na cílovou identitu. Architektura se skládá ze dvou enkodérů a mapovací sítě, která mapuje oba vstupy do latentního prostoru sítě StyleGAN2. Ta následně vygeneruje výsledný obrázek ve vysoké kvalitě. Trénování je *self-supervised* bez potřeby označených dat. Naše metoda dokáže generovat obrázky téměř v reálném čase a umožňuje syntézu náhodných identit s nezávisle ovladatelnou pózou a výrazem.

Klíčová slova generování lidských obličejů, přenos výrazu, přenos pózy, StyleGAN2, hluboké učení

Contents

List of Figures

List of Tables

Introduction

Animating facial portraits in a realistic and controllable manner has numerous applications in image editing as well as interactive systems. For instance, a natural-looking animation of an on-screen character with various human poses and expressions can enhance the user experience in games or virtual reality applications. Achieving this goal is a challenging task, as it requires representing the face (e.g. modeling in 3D) in order to control it and developing a method to map the desired form of control back onto the face representation. The form of control can be another face portrait. More specifically, in the task of pose and expression transfer, another face portrait is used to drive the target image which is the case for this thesis.

With the advent of generative models, it has become increasingly easier to generate high-resolution human faces that are virtually indistinguishable from real images. StyleGAN2 [1] achieves the state-of-the-art level of image generation with high quality and diversity among GANs (Generative Adversarial Networks) [2].

StyleGAN2 generates human faces by inputting a latent code, which is a vector sampled usually from Gaussian distribution, to the generator. We can semantically edit the images in the latent space, enabling us to change age, gender, smile, and other features. One common technique to do that is to identify linear semantic directions in the latent space and edit images by manipulating the latent code in these directions. However, these linear semantic directions are entangled, resulting in unwanted secondary edits (e.g. generating a person from a different viewpoint might make them grow a beard, age, change hairstyle, or change identity completely).

Nevertheless, the generated images are still random, and we want to edit images of real people. GAN inversion aims to reconstruct an image of a real person by finding a latent vector that best represents the target image when sent through the generator. When the corresponding latent code is found, the aforementioned method can be used for editing. However, it still suffers from the same shortcomings. That is why we take a look at non-linear edits in the latent space of StyleGAN2 which we believe have not been broadly studied.

This work aims to develop a method that enables the synthesis of a new image of an individual by taking both a driving image and an identity image as input, incorporating the pose and expression of the person in the driving image into the generated image from the identity image. Our method is self-supervised, and it does not require labeled data. Additionally, it fully relies on neural rendering in a one-shot setting without using a 3D graphics model of the human face. By eliminating the need for a 3D graphics model and labeled data, our approach provides a more efficient and practical solution for face synthesis. We review existing methods and evaluate our approach on pose and expression transfer fidelity as well as on face identity preservation.

The rest of the thesis is structured as follows. Chapter 1 presents related work regarding human face modeling, generation, and facial pose and expression transfer. Chapter 2 discusses the technical background of StyleGAN and its latent space manipulation. The architecture, dataset, and training details are described in Chapter 3. A baseline method derived from the linear semantic directions is presented in Chapter 4, along with a comparison and qualitative and quantitative evaluation of our approach.

We expect the reader to have a basic understanding of fundamental Machine learning concepts as they will not be discussed in this work. One may find a comprehensive introduction to Machine learning at e.g [3, 4].

Chapter 1

Related Work

There has been a tremendous amount of research put into human face modeling/generation, face manipulation, and motion (e.g. pose, expression) transfer. In this chapter, we will go over the most relevant and influential works in the field.

1.1 Parametric modeling of faces for image generation

Traditionally, facial animation given an image was performed by fitting a statistical model such as AAM (Active Appearance Model) [5] or 3DMM (3D Morphable Model) [6]. To construct a 3DMM, a set of 3D face scans is first aligned and normalized to remove any variations in pose, expression, and scale. The aligned scans are then used to construct the shape and texture models, which capture the statistical variations of the 3D facial shape and texture, respectively. These models are typically represented using a low-dimensional subspace spanned by the principal components of their corresponding variations. By fitting these models to a single input image, facial animation can be achieved by modifying the estimated parameters with a certain degree of accuracy. Active appearance models work in a similar way except instead of using 3D face scans, a set of 2D images are aligned based on facial landmarks (e.g. the corners of the eyes, the tip of the nose, and the corners of the mouth).

Many works build on top of these statistical models to improve the pose and expression transfer. MLM (Multilinear model) [7] is an extension of 3DMM. MLM represents the facial shape and texture variations as a tensor, which captures the correlations between different modes of variations. This means that MLM is able to capture better the complex variations in facial features, including the relationship between different features. Work [8] utilizes 3DMM to estimate the parameters that correspond to the facial expressions of the source actor and applies them to the target actor. The

method involves first tracking the facial landmarks of both actors and then using these landmarks to compute the expression parameters of the source actor. The estimated parameters are then transferred to the target actor by warping the target's face based on the correspondences between the source and target landmarks. Face2Face [9] builds on top of this work by utilizing a blendshape model to better model and transfer the facial expressions.

For an in-depth overview of parametric face models and their possible applications we refer to a survey paper [10].

1.2 Deep Learning-based approaches for image generation

Many works achieve remarkably good results with the aforementioned statistical models combined with deep learning [11, 12, 13] (just to name a few), however, these works are not as relevant to our work as the ones which drop the parametric representation of the face.

Supervised approaches for face control learn to model factors of variation such as lighting and pose by conditioning the generated image on known ground truth information which may be head pose, expression, or landmarks [14, 15, 16]. This requires a training dataset with known pose or expression information that may be expensive to obtain. These datasets often have a very limited set of expressions (e.g. smile, frown, neutral, etc.).

That is why unsupervised and self-supervised methods have become increasingly popular in this domain [17, 18, 19, 20]. The so-called Deepfake was first developed by a Reddit user using an autoencoder-decoder pairing structure. The autoencoder extracts latent features of face images and the decoder is used to reconstruct the face images. To swap faces between source images and target images, there is a need for two encoder-decoder pairs. Given two sets of images from two different identities, each pair is trained to reconstruct images from their corresponding set. However, the two pairs share the same encoder network which enables the encoder to learn all of the mutual features (e.g. pose and expression), while the decoder is trained to learn person-specific features. To create a deepfake a source image of the first person is encoded with the common encoder and decoded with the decoder that was trained to reconstruct the second person [21].

CycleGAN [22] is another self-supervised method that can be used to transform images from one domain into another. Although it was not originally developed for this specific task, CycleGAN is trained to be cycle-consistent, which means that the generated images often retain some semantic similarities to the original images. For instance, if a CycleGAN model is trained to transform images of one person's face (domain A) into those of another person (domain B), it may learn to map the pose, position, or expression of the face in domain A onto the generated face in domain B [17]. However, both of the

aforementioned methods can only represent the single identity on which the model was trained. Recent works aim to develop models that can generate images for any identity, even those that were not present in the training data."

X2Face [17] demonstrates that an encoder-decoder architecture with a large collection of video data can be trained to synthesize human faces conditioned by a source frame without any parametric representation of the face or supervision. Furthermore, the paper shows that the expression can be driven not only by the source frame but also by audio to some degree of accuracy. Similarly, paper [18] employs a GAN architecture with an added embedding network that maps face images with estimated facial landmarks into an embedding that controls the generator. This allows for conditioning the generated image only on facial landmarks.

The approach proposed in [19] enables the generation of a talking-head video from a single input frame and a sequence of unsupervisedly-learned 3D keypoints that represent the motions in the video. By utilizing this keypoint representation, the method can efficiently recreate video conference calls. Moreover, the method allows for the extraction of 3D keypoints from a different video, enabling cross-identity motion transfer.

Recently, Megaportraits [20] have achieved a state-of-the-art level of one-shot cross-reenactment quality. Their method utilizes an appearance encoder, which encodes the source image into a 4D volumetric tensor and a global latent vector, and a motion encoder which extracts motion features from both of the input images. These features together with the global latent vector predict two 3D warpings. The first warping removes the source motion from the volumetric features, and the second one imposes target motion. The volumetric features are processed by a 3D generator network and together with the target motion are input to a 2D convolutional generator that outputs the final image.

1.3 Face manipulation with StyleGAN

StyleGAN generates human faces by inputting a latent code into the generator. Its architecture and the overall latent space manipulation will be closely described in the next chapter 2. Works that focus on StyleGANs ability to condition the generated image based on pose and expressions will be described in this chapter.

GANSpace [23] analyzes the latent space of StyleGAN and creates interpretable controls for image synthesis, such as pose, lighting, and simple expressions. The important latent directions are identified based on PCA that is applied in the latent space. However, these latent directions are heavily entangled, meaning that one learned latent direction might influence other facial attributes as well. For example, given a learned latent direction of a pose change, when applied, the person might grow a beard, change hairstyle or even change identity. Similarly, InterFaceGAN [24] introduces a framework

for interpreting the disentangled face representation by aligning the learned semantic directions with a set of annotated facial attributes. This is done by utilizing a pre-trained classifier to predict various facial attributes, such as pose, gender, and smile. The classifier is then used to identify the directions in the disentangled space that are most predictive of each facial attribute. The paper then demonstrates that this method better preserves identity compared to PCA baseline. However, since the classifier is trained to predict a very limited set of facial attributes, the resulting semantic directions only capture a small subset of the possible facial variations. StyleFlow [25] achieves better results by modeling StyleGANs latent space through a series of continuous, invertible transformations, conditioned on input image attributes. This allows for greater flexibility in manipulating image attributes and results in smoother transitions between images.

Rather than relying on the linear semantic latent directions of StyleGAN, StyleRig [26] leverages a 3D Morphable Model (3DMM) to control the semantic parameters of the generated images. This is achieved by utilizing a pre-trained face reconstruction network that maps a latent code of the source image to a vector of semantic control parameters of the 3DMM. Additionally, an encoder is used to map the latent code of the target image to a lower-dimensional vector, which is then combined with the vector of control parameters via a decoder. The resulting latent code is fed into the StyleGAN generator to produce the final output image.

Chapter 2

Technical Background

In this chapter, we provide an overview of the technical concepts and tools that are essential for understanding the research presented in this thesis. Firstly, we introduce Generative Adversarial Networks. In particular, we focus on StyleGAN as it is utilized in our work. Lastly, we discuss the latent space and its observed potential for image editing.

2.1 Generative Adversarial Networks

GAN (Generative Adversarial Network) [2] is a framework introduced by Goodfellow et al. in 2014 consisting of a discriminator and a generator. The generator tries to match the distribution of the training dataset while the discriminator tries to predict which image is generated and which is from the original dataset. The training procedure for the generator is to maximize the probability of the discriminator making a mistake. This corresponds to a two-player minimax game. Such training process can be very unstable and often suffers from many issues (e.g. mode collapse or vanishing gradients). Mode collapse occurs when the generator learns to produce only a limited set of outputs of the target distribution. Vanishing gradients occur when the discriminator learns too fast compared to the generator and in that case, the generator's gradients approach zero, effectively making the generator unable to learn anything from the discriminator's feedback. These issues were discussed in [27] where DCGAN (Deep Convolutional Generative Adversarial Network) was introduced and several techniques were proposed to mitigate these problems, including adding noise to the input and using mini-batch discrimination. Another influential work trying to overcome these issues was Wasserstein GAN [28] which introduced a new objective function based on Wasserstein distance.

DCGANs were the state of the art among the GANs in image generation before the introduction of Progressive Growing GANs in 2017 [29]. The idea behind Progressive GANs is that the discriminator and the generator are

trained progressively on increasingly higher resolutions of images. This helps stabilize the training procedure and leads to significant improvements in the quality of generated images because it allows the generator to capture even finer details.

2.2 StyleGAN

Traditional Progressive GANs provide the latent code directly into the input layer (Figure 2.1a). StyleGAN, which was introduced in 2018, deviates from this approach by excluding the input layer and instead initiates the model from a learned constant [30]. Given a latent code $\mathbf{z}$ in the latent space $\mathcal{Z}$ a non-linear mapping network $f : \mathcal{Z} \to \mathcal{W}$ first produces $\mathbf{w} \in \mathcal{W}$ (Figure 2.1b left). The $\mathbf{w}$ then gets replicated to each layer and through the learned affine transformation produces styles. The styles control the AdaIN (adaptive instance normalization), which adapts the mean and standard deviation of content feature maps to match those of style feature maps [31].

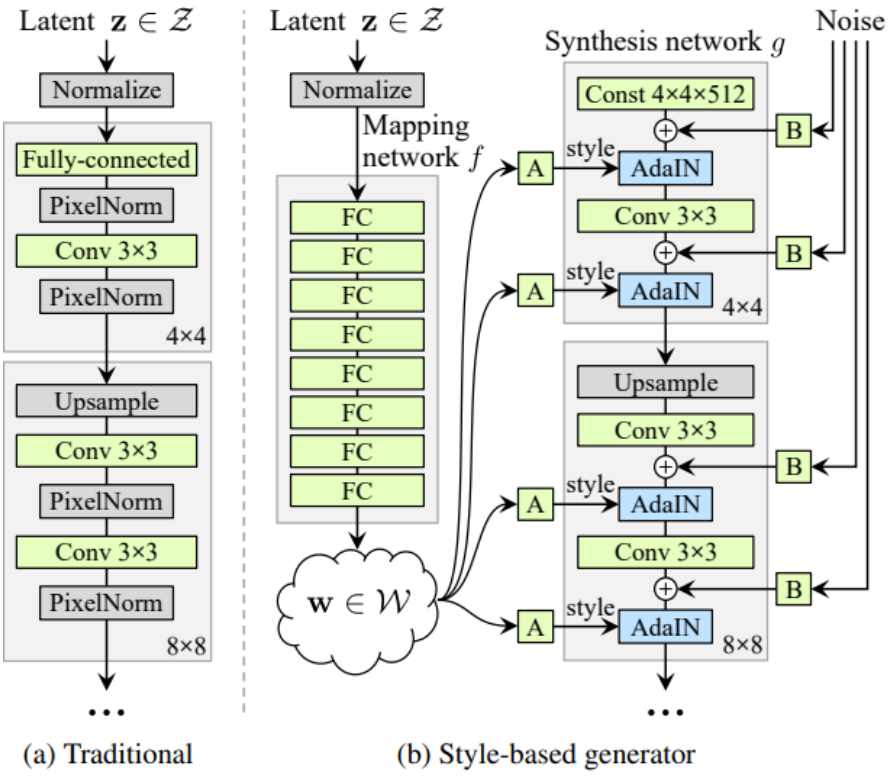

Figure 2.1: Comparison of a traditional Progressive GAN architecture (a) and StyleGAN architecture (b). The style codes control the AdaIN added after each convolution. Noise is injected to generate stochastic variation. "B" denotes learned per-channel scaling [30].

The synthesis network (Figure 2.1b right) consists of 18 layers, with two layers dedicated to each resolution (4^2 – 1024^2). Styles in coarse spatial layers corresponding to resolutions (4^2 – 8^2) bring high-level attributes such as pose, general hairstyle, face shape, and glasses. Styles in middle layers associated with resolutions (16^2 – 32^2) are responsible for smaller-scale facial features, hairstyle, and whether the eyes should be open or closed. The final layers which correspond to resolutions (64^2 – 1024^2) bring mainly the color scheme and microstructure. Simultaneously noise is being input at each layer before the AdaIN operation to ensure some variation in the images. The noise affects only inconsequential stochastic variation such as differently combed hair, skin pores, and freckles. It also makes the images look more realistic as without the noise input the images have a "painterly" look [30].

StyleGAN was trained on the FFHQ (Flickr-Faces-HQ) dataset which consists of more than 70 000 high-quality face portraits crawled from Flickr. The images have been preprocessed by filtering low-quality images and removing unwanted backgrounds. The faces have also been cropped and aligned, removing undesirable extreme head poses [30].

2.3 Latent space manipulation

Regarding image editing in the latent space of GANs, paper [27] first noticed the arithmetic properties of the generator's latent space. They demonstrate this effect on many examples such as taking a latent code of a man with glasses, subtracting a latent code of a man without glasses, and adding a latent code of a woman without glasses produces a woman with glasses. Since then, researchers have extensively studied both the linear properties and non-linear edits that can be made in the generator's latent space.

Specifically for StyleGAN, a tremendous amount of work has been published regarding the latent space exploration [23, 24, 25, 32, 33]. InterFaceGAN [24] shows that linear semantic directions can be easily found in a supervised manner. First, a large collection of images are synthesized by randomly sampling from the latent space. Then, a pre-trained attribute prediction model is used to assign attribute scores for all of the images which are then used to train a linear SVM. This produces a decision boundary – a hyperplane whose normal vector is the semantic latent direction of the predicted attribute. Figure 2.2 demonstrates a learned semantic direction of yaw change. The latent directions are heavily entangled, meaning that one learned latent direction will likely influence other facial attributes as well. For example, given a learned latent direction of a pose change, when applied, the person might change expression, hairstyle, or even identity.

These are still synthesized images. If we want to generate and manipulate a particular person, we first have to invert the given image which is discussed in the next section 2.4.

Figure 2.2: Learned semantic direction of yaw applied over three random latent vectors. Due to the entanglement of the directions, the pose change influences the expression, gender, age and identity of a person. With the increased magnitude of the semantic direction (in this case yaw direction), artifacts start to emerge [33].

2.4 GAN Inversion

The process of finding a latent code that can generate a given image is referred to as the task of image inversion [34, 35, 36]. There are two latent spaces considered for this task. The native StyleGAN latent space $\mathcal{W}$ where a given 512-dimensional latent code $\mathbf{w}$ is shared across all of the generator's layers. The other one is the extended latent space where each layer is considered separately, resulting in a larger extended latent space $\mathcal{W}^+$ of 18×512 dimensions. It has been shown that for the purpose of inverting an image the extended latent space produces better results [37].

There are mainly two approaches to image inversion. Either through direct optimization of the latent code to produce the specified image [32, 37, 38, 39] or through training an encoder on a large collection of images [40, 41, 42]. Typically, direct optimization gives better results, but encoders are faster. Additionally, encoders display a smoother behavior, producing more coherent results over similar inputs [43]. We take advantage of this in our work.

It has been demonstrated [39, 42] that in comparison with $\mathcal{W}^+$, $\mathcal{W}$ provides a higher degree of editability, meaning latent codes in this space can be more easily manipulated while maintaining a greater level of realism. However, $\mathcal{W}$ has poor expressiveness, resulting in inversions that are often inconsistent with target identity. Therefore there exists the so-called distortion-editability trade-off [39]. Recently paper [38] has shown that this trade-off can be bypassed by using PTI (Pivotal Tuning Inversion). The idea is that one may fine-tune the generator around an initial latent code called the pivot. This

achieves state-of-the-art inversion and a high level of editability. However, this approach requires storing corresponding generator weights for each individual inversion.

In our work, we opted to use an encoder for the sake of image inversion, as we require many training images to be inverted and direct optimization of each training sample would not be computationally feasible. We chose ReStyle [41] which uses an encoder in an iterative fashion to refine the initial estimate of the latent code. This approach is a suitable fit for our purpose as it leverages smoother behavior over similar inputs from encoders as well as better reconstruction quality from iterative optimization. Currently, encoders supported in ReStyle are pSp (pixel2style2pixel) [40] and e4e (encoder4editing) [42]. While both of these encoders embed images into the extended latent space $\mathcal{W}^+$, Tov et al. [42] argue that by designing an encoder that predicts codes in $\mathcal{W}^+$ which reside close to $\mathcal{W}$ they can better balance the distortion-editability trade-off. However, we chose to use ReStyle with a pSp encoder in our network as the baseline method with e4e encoder – discussed in section 4.1 – had trouble preserving the target identity.

Chapter 3

Method

Our framework takes two face images as input, a source (driving) face image, and a target (identity) face image. The network produces an output image where the pose and expression from the source face image are transferred onto the target identity.

3.1 Architecture

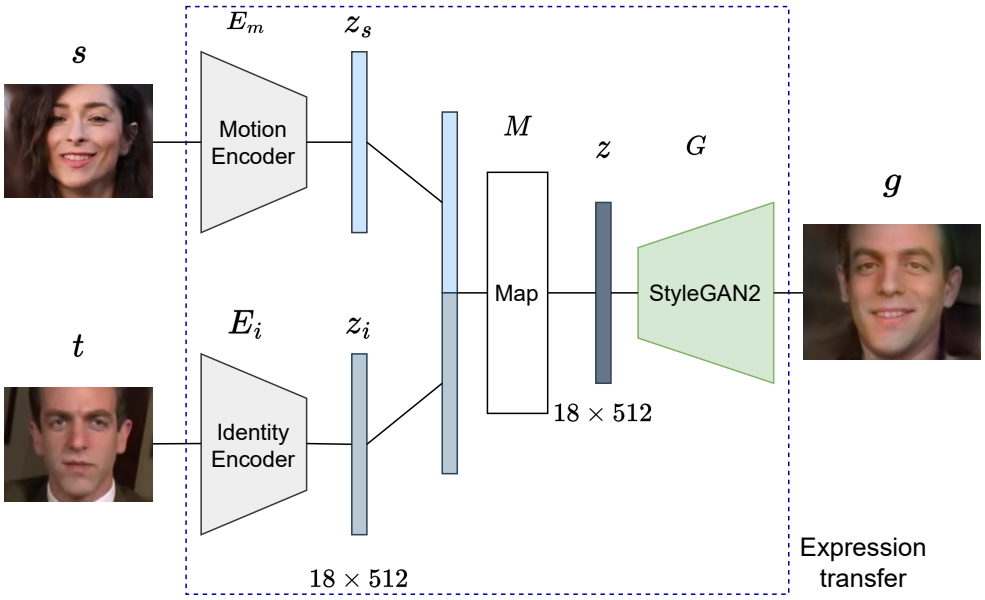

Figure 3.1: The architecture of the proposed model. The Motion encoder and Mapping network weights are being optimized, while the Identity encoder and StyleGAN2 weights stay fixed during training.

Figure 3.1 depicts the proposed architecture. The network consists of a motion encoder E_m, an identity encoder E_i, a mapping network M, and a pre-trained generator network G. The encoder E_i embeds the identity of the target face image. The encoder E_m embeds motion – the pose and expression of the source face image. The mapping network then projects the outputs of the encoders into the latent space of the pre-trained StyleGAN2 generator. This approach offers the advantage of generating high-quality images through StyleGAN while avoiding the intricate training process of GANs. The network architecture is inspired by [44].

Specifically, a source image s and a target image t are aligned and resized to 256×256 pixels and then fed into their corresponding encoders, where they get embedded into the extended latent space $\mathcal{W}^+$ of 18×512 dimensions. Embeddings z_s for pose and expression of source image s and z_t for the identity of target image t are then concatenated and transformed through the mapping network into a latent code $z \in \mathcal{W}^+$ that is then used as an input for the generator that finally produces an output image g. The image generation can be formally expressed as

$$g = G\Big(M\Big(E_m(s) \ \oplus \ E_i(t)\Big)\Big),$$

where $\oplus$ denotes concatenation.

It has been shown that network architecture ResNet-IR SE 50 can be trained to embed various entities into the latent space of StyleGAN2 such as cartoons [40], hair [44] and much more. That is why we utilize this network for the E_m encoder and we adapt it from the repository published alongside the pSp paper [40]. For encoder E_i, we use a pre-trained ReStyle with the pSp configuration as discussed in section 2.3. As for the Mapping network M, we employ one fully connected linear layer. For the generator, we use the pre-trained StyleGAN2 that outputs high-resolution images of 1024×1024 pixels.

3.2 Training

We employ self-supervised training to optimize the parameters of the encoder E_m and the mapping network M, while keeping the parameters of the generator G and the encoder E_i fixed. The training is performed on an unlabeled dataset of short video clips, each containing a single person. Figure 3.2 illustrates the training procedure.

During each iteration of the training procedure we randomly sample two pairs (s, t) of frames from two video clips. More specifically, a source and target frame pair (s_A, t_A) of identity A are randomly sampled from a video clip and another source and target frame pair (s_R, t_R) of a random identity are randomly sampled from a differnt video. We then generate two images

$g_{s_A \to t_A}$ where the source and target frames are from identity A and $g_{s_A \to t_R}$ where the source frame is of identity A and the target frame is of identity R. The notation $g_{s_A \to t_R}$ implies that the pose and expression from the source image s_A is imposed onto the indentity R from the target image t_R. We employ the following loss functions:

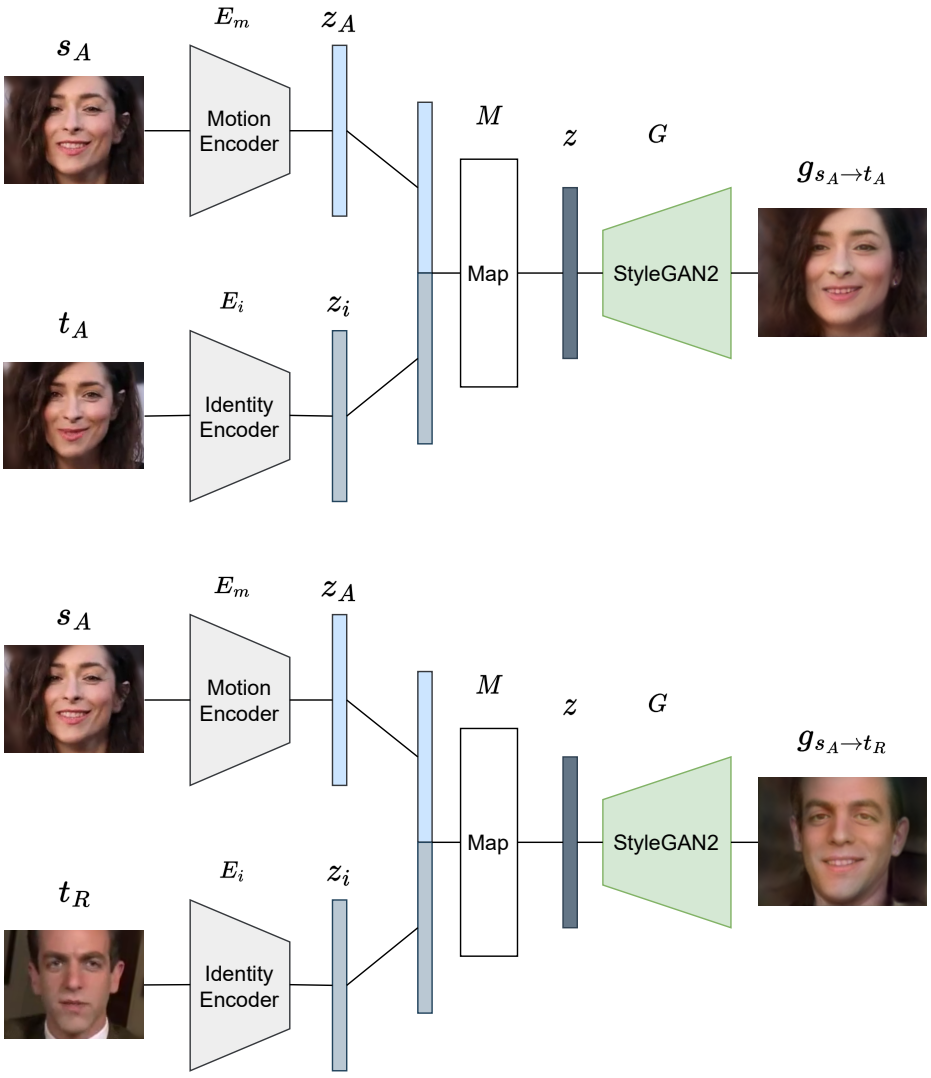

Figure 3.2: Training procedure of the proposed method. Two pairs of frames are sampled from two videos that are then used to generate one self-reenactment image and one cross-reenactment image.

Pixel-wise loss. Pixel-wise loss or more specifically $\mathcal{L}_2$ loss simply measures the squared distance of corresponding pixels of the two images. Formally this can be expressed as

$$\mathcal{L}_2 = \|s_A - g_{s_A \to t_A}\|_2,$$

where s_A is the source frame of identity A and $g_{s_A \to t_A}$ is a generated image using both inputs from identity A.

Perceptual loss. The pixel-wise loss falls short in capturing the perceptual changes that humans notice when viewing images. This can be demonstrated on blurring an image, which may not cause a significant change in the $\mathcal{L}_2$ value but can still be visually noticeable to a human eye. To address this limitation, the perceptual loss has been developed. Instead of analyzing individual pixels, perceptual loss compares the higher-level similarities between two images, such as their content and style. LPIPS (Learned Perceptual Image Patch Similarity) loss has been proposed in [45], which involves utilizing a dataset of almost half a million human judgments to compute the perceptual distance between reference and generated images. This approach improves upon the limitations of the pixel-wise function and better captures the perceptual differences that humans notice when viewing images. We employ a pre-trained neural network to calculate the perceptual distance

$$\mathcal{L}_{LPIPS} = 1 - \langle P(s_A), P(g_{s_A \to t_A}) \rangle,$$

where P is a perceptual feature extractor that outputs unit-length normalized features and $\langle\rangle$ denotes dot product. We again use only images of the same identity in this loss.

Identity loss. Using only the aforementioned losses worked well in the self-reenactment scenario, where the source and target frames are from the same video. However, in the cross-reenactment scenario, where the source and target frames contain different identities, fails miserably. That is why we employ the pre-trained state-of-the-art facial recognition model ArcFace which utilizes the Additive Angular Margin Loss [46]. We calculate it in a similar fashion as the previous loss:

$$\mathcal{L}_{ID} = 1 - \langle D(g_{s_A}), D(g_{s_A \to t_R}) \rangle,$$

where D produces unit-length normalized embeddings, g_{s_A} is the inverted source frame of identity A (Image generated with the latent code produced by E_i from s_A), and $g_{s_A \to t_R}$ is a generated image where the source frame is of identity A and the target frame is of identity R.

CosFace loss. Finally, we implement the CosFace loss [47] which we use in a similar manner as in the Megaportraits paper [20]. For this loss only motion descriptors, that are embedded by E_m, are necessary. We separately

calculate a motion descriptor $z_R = E_m(s_R)$ while also storing the motion descriptor calculated during the forward pass of the network — $z_A = E_m(s_A)$. Lastly, the motion descriptors of the generated images $z_{A\to A} = E_m(g_{s_A\to t_A})$ and $z_{A\to R} = E_m(g_{s_A\to t_R})$ are calculated. We then arrange them into positive pairs P that should align with each other: $P = (z_{A\to A}, z_A), (z_{A\to R}, z_A)$, and negative pairs: $N = (z_{A\to A}, z_R), (z_{A\to R}, z_R)$. These pairs are then used to calculate the following cosine distance:

$$d(z_i, z_j) = a \cdot (\langle z_i, z_j \rangle - b),$$

where both a and b are hyperparameters. This distance is then used to calculate the Large Margin Cosine Loss [47]:

$$\mathcal{L}_{cos} = -\sum_{(z_k,z_l)\in\mathcal{P}} \log \frac{\exp\{d(z_k, z_l)\}}{\exp\{d(z_k, z_l)\} + \sum\limits_{(z_i,z_j)\in\mathcal{N}} \exp\{d(z_i, z_j)\}}.$$

The idea behind this loss is that when we randomly sample a source frame from each of the two different videos, the pose and expression in these frames will likely differ. In that case, the generated image, using one source frame, should have a similar motion descriptor to that particular source frame while also having a dissimilar motion descriptor to the other source frame regardless of the identity used to generate the image.

We also use crop versions of $\mathcal{L}_2$ loss and $\mathcal{L}_{LPIPS}$ loss where we crop the inner part of the face from the aligned image. The resulting cropped face image is of 188×188 pixels from the original 256×256 pixels of the aligned image. The losses $\mathcal{L}_{2\_crop}$ and $\mathcal{L}_{LPIPS\_crop}$ are used exactly as their aforementioned counterparts.

To conclude, the total loss which is used to train the network is the weighted sum of the individual losses:

$$\begin{aligned}\mathcal{L} = w_{\mathcal{L}_2}\mathcal{L}_2 + w_{LPIPS}\mathcal{L}_{LPIPS} + w_{ID}\mathcal{L}_{ID} + w_{cos}\mathcal{L}_{cos} \\ + w_{\mathcal{L}_{2\_crop}}\mathcal{L}_{2\_crop} + w_{LPIPS\_crop}\mathcal{L}_{LPIPS\_crop}\end{aligned}$$

3.3 Dataset and data augmentation

For our goal, we need a large dataset consisting of numerous unique identities and a wide range of images with varying poses and facial expressions for each identity. To meet this requirement, it was necessary to resort to using video data despite the potential trade-off in image quality.

We decided to use the VoxCeleb2 dataset [48] which was created in 2018 originally for speaker recognition and verification. It has since been used for talking head synthesis, speech separation as well as face generation. It contains over a million utterances from 6 112 identities, providing us with a

vast array of subjects to work with. The dataset is primarily composed of celebrity interview videos, offering a broad spectrum of poses and expressions to utilize. The videos are categorized by identity, and trimmed into shorter utterances that range from 5 to 15 seconds in duration. They have also already undergone preprocessing that includes cropping the frames to the bounding boxes around each speaker's face.

Unfortunately the preprocessing step does not match the one required by StyleGAN as the faces are cropped by the forehead. StyleGAN requires the images to be of the entire head including the top part that is missing. We use the official preprocessing script provided by StyleGAN that pads the missing part of the forehead. This results in all of our training and testing images having blurred stripes at the top. Although this hinders the quality of the inverted images as well as the generated ones, it is better than the alternative datasets that we considered. The alternatives did not have a large collection of speakers and they were not segregated based on identity, resulting in multiple individuals speaking in a single utterance.

Another challenge with the dataset is the relatively low resolution of the videos, typically 224×224 pixels. This is problematic since StyleGAN is designed to generate high-quality images with a resolution of 1024×1024 pixels. Nevertheless, finding a large dataset of high-resolution videos featuring a vast number of distinct individuals is nearly impossible.

As the number of videos per individual differs quite drastically, we tried to balance it out by only using a maximum number of videos per person. We extracted 10 frames at half-second intervals from each video. Subsequently, we filtered the sampled images to eliminate the ones with extreme poses, aligning them with the image filtering approach implemented in the FFHQ dataset, which served as the training set for StyleGAN. We also pre-align all of the images using the official StyleGAN preprocessing script, which uses dlib [49], a machine-learning library, to detect human faces and Facial Landmarks. If the image has such a bad quality that the face or the Facial Landmarks are not detected, the image is dropped as well. The filtered training dataset contains around 6 000 different identities each with around 10 images from 5 different video clips, resulting in a little under 300 000 images.

3.4 Training and implementation details

The model, as well as the dataset, is very large and thus we trained it for a million steps with a batch size of 8. Surprisingly, even after that many steps, the validation loss kept slowly decreasing. However, we observed a decline in the ability of the model to accurately capture facial expressions in the generated images around the one million training step mark, despite the decreasing validation loss. The loss function does not capture the expression perfectly and that is the reason why we had to revisit all of the intermediate

model weights and check if there are better ones even with higher validation loss. We chose the model weights which transferred the expression the best on the validation set based on the non-differentiable evaluations discussed in 4.4.

We used the ranger optimizer [50], which combines Rectified Adam algorithm and Look Ahead. We set the learning rate to $1 \cdot 10^{-5}$. For our model with the best performance, we used the following hyperparameters for the losses: $w_{\mathcal{L}_2} = 0$, $w_{LPIPS} = 0.05$, $w_{ID} = 0.3$, $w_{cos} = 0$, $w_{\mathcal{L}_{2\_crop}} = 2$, $w_{LPIPS\_crop} = 0.3$. We also set the parameters $a = 5$ and $b = 0.2$ in the CosFace loss. The parameters of our best model heavily rely on crop versions of the self-reenactment losses. The reason for this is that when we tried to use full images as input for those losses, the network struggled to learn the desired facial expression manipulation. It instead had to focus on the background and hair fidelity and thus failed to transfer the expressions correctly.

Chapter 4

Experiments

This chapter presents experiments and evaluations of our proposed method. We begin by introducing the baseline method and then describe the variants of our method. We compare the results of our approach with the baseline methods and perform both qualitative and quantitative evaluations to demonstrate the pose and expression transfer fidelity as well as identity preservation.

4.1 Baseline method

For our baseline method, we take advantage of the arithmetic property of the StyleGANs latent space. As mentioned before, the latent space has a linear property, where the latent codes can be added and subtracted for meaningful edits. However, these edits have difficulties preserving the identity of the person.

Given two frames A_0 and A_1 (sampled from the same video) where the pose and expression of the person differ, the edit vector is represented by the difference between the latent codes corresponding to the inverted frames. The pose and expression can then be imposed onto a random person in image R by adding this latent code of edit to a latent code corresponding to the image R. Formally this can be written as

$$z_{A_1 \to R} = z_R + \alpha \cdot (z_{A_1} - z_{A_0}),$$

where z_R is the latent code of a random person, z_{A_0} is the latent code of the person A with the initial pose and expression and z_{A_1} is the latent code of the same person with a different pose and expression. The α represents the magnitude of the edit and the latent code $z_{A_1 \to R}$ when fed into the StyleGAN generates a person R with the pose and expression from A_1. In our case, we always set the α to one, because we want the same expression and pose.

However, this approach requires the initial pose and facial expression in frame A_0 to match the pose and expression of the person in frame R. This is

a very strict requirement as there likely will not be a frame in a video, where the pose and expression match perfectly the pose and expression in frame R. Even when two short videos are considered, there might not be two frames, each from a different video, where the pose and expression match precisely.

Instead of searching for two frames that match pose and expression the best, we utilize the arithmetic property again. We flip each frame in a video by the vertical axis and invert them along with their non-flipped counterparts. Then, we calculate the mean latent code across all of the frames. This results in a frontal pose with an average expression across the video – typically a neutral expression. We do this for both of the videos which provides us with the same pose and a similar expression for the initial frames. We then used the aforementioned method to transfer pose and expression from one person to another. The downside of this method is that it does not work with single images but rather requires a short video of each individual. It also requires inverting all of the frames within the videos.

We consider two versions of the baseline method. Both of them invert all of the images with ReStyle [41], but one with the pSp encoder [40] configuration and the other with the e4e [42] as discussed in detail in section 2.4.

4.2 Experiments with our method

Apart from using the already mentioned parameter configuration 3.4 we also tried utilizing the CosFace loss function, but unfortunately for our architecture, this slowed down the loss calculation a lot without much-added benefit. To calculate this loss it is necessary to calculate 3 additional motion descriptors, which means 3 additional forward passes of a large encoder network E_m.

Another thing we tried with this architecture was optimizing the pretrained generator weights, a similar idea to Pivotal Tuning [38]. The idea is that by optimizing more parameters, the network could learn more complex facial expressions, which were proving difficult for the current model.

4.3 Qualitative evaluation

We start by showing the more straightforward case – a self-reenactment scenario, where the source and target image are from the same identity. Figure 4.1 shows that the pose and expression transfer between the same identities works very well, but the network still struggles with the hairstyle transfer due to reasons discussed in section 3.4. In the rest of the qualitative and quantitative evaluations, we focus on the cross-reenactment scenario, where the input identities differ, which is the main focus of the research.

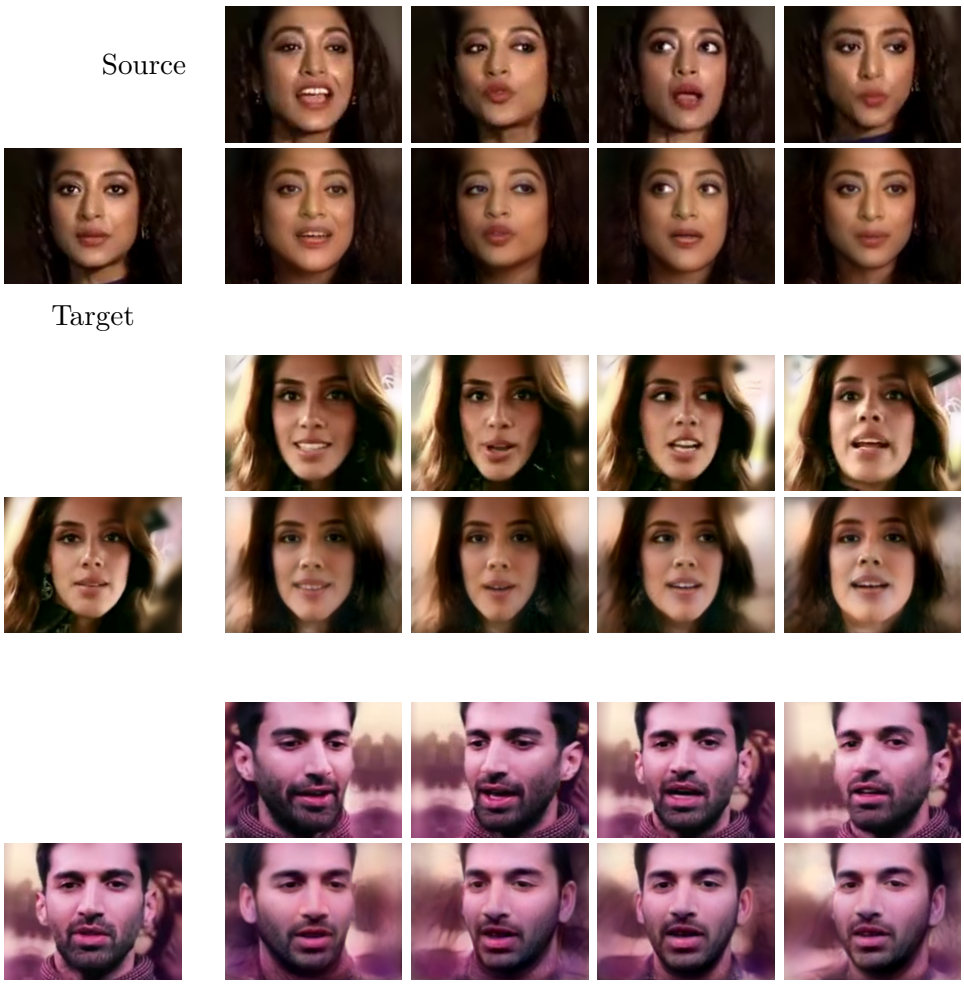

Figure 4.1: Examples of pose and expression transfer for three distinct identities in a self-reenactment scenario, where both the target and source images belong to the same identity. In the leftmost column, the target image is displayed. The first row shows the source images while the second row presents the corresponding generated images.

Although our method works with individual images in a one-shot setting it can be used to produce videos. The performance can be better observed in a video format where the range of the pose transfer as well as the expression transfer can be better appreciated than from static images. A few videos are included in the attachments.

In figure 4.2 we present several examples of pose and expression transfer between a variety of identities. The pairs are challenging since they vary in ethnicity, gender, and illumination. Another challenge is accessories that people wear such as eyeglasses or earrings.

Figure 4.2: Pose and expression transfer examples. The Top box depicts the identity input image along with their inversion. The bottom grid shows the transfer of pose, expression, and eye movement to different identities. The identities are preserved column-wise, and the poses and expressions are preserved row-wise.

The pose and expression are transferred while still preserving the input identity. The model was able to train to transfer pose, expression, and eye movement. The network also correctly identifies, that if eyeglasses are present in the identity image, they should be preserved in the output image. Surprisingly, the network is able to model eye movement even behind the eyeglasses. The network fails to preserve hair or background correctly as discussed in 3.4.

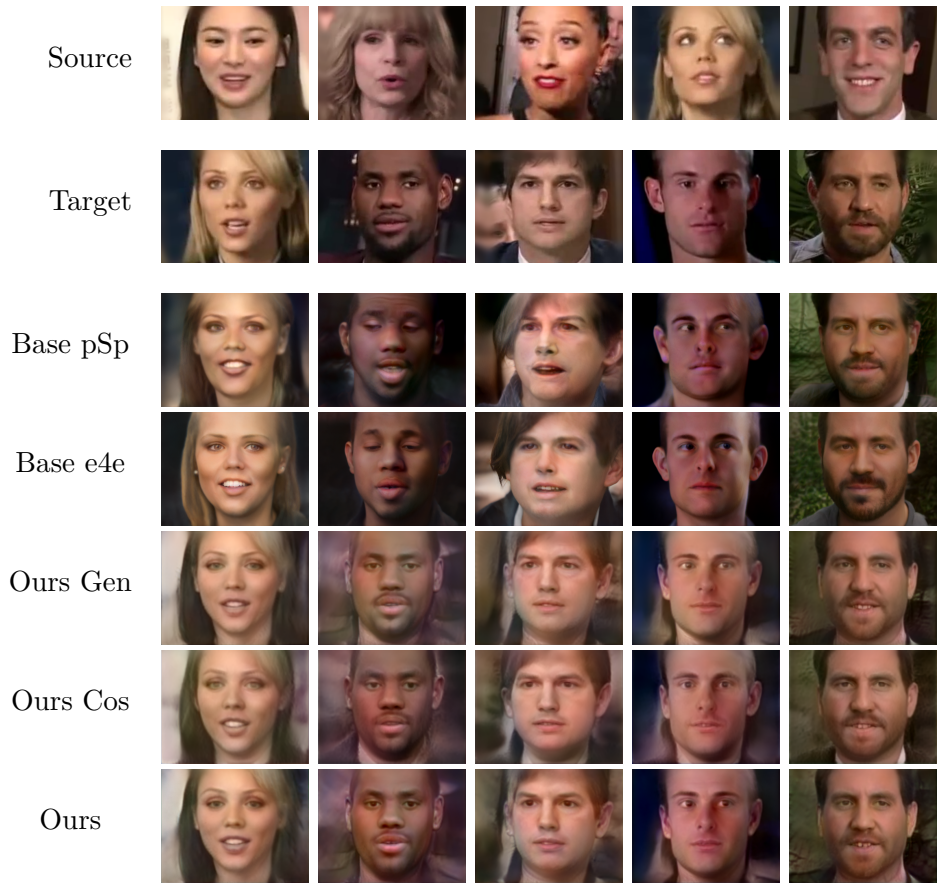

Figure 4.3: Pose and expression transfer comparison. The top two rows represent the input source and target images. The next two rows show the baseline method results, first with pSp inversion configuration and second with e4e. The last three rows depict the results of our method, the first with generator weight optimization, the second with utilizing the CosFace loss, and the last shows the best model.

In Figure 4.3, we compare the results of the baseline method with the variants of our proposed method. The baseline method does not use the target image, but rather a frontal representation with an average expression across the video of the identity as explained in section 4.1. The Figure shows that the baseline method has trouble preserving the identity of the target person and some artifacts are present. Some expressions are transferred more faithfully compared to our method. However, it can happen, that the average expression in one video is not the same as in the other, and then the expressions are not transferred correctly. This can be seen in the second and last columns of the figure 4.3. Our best model represents eye movement better compared to our other variants while also generating more realistic images.

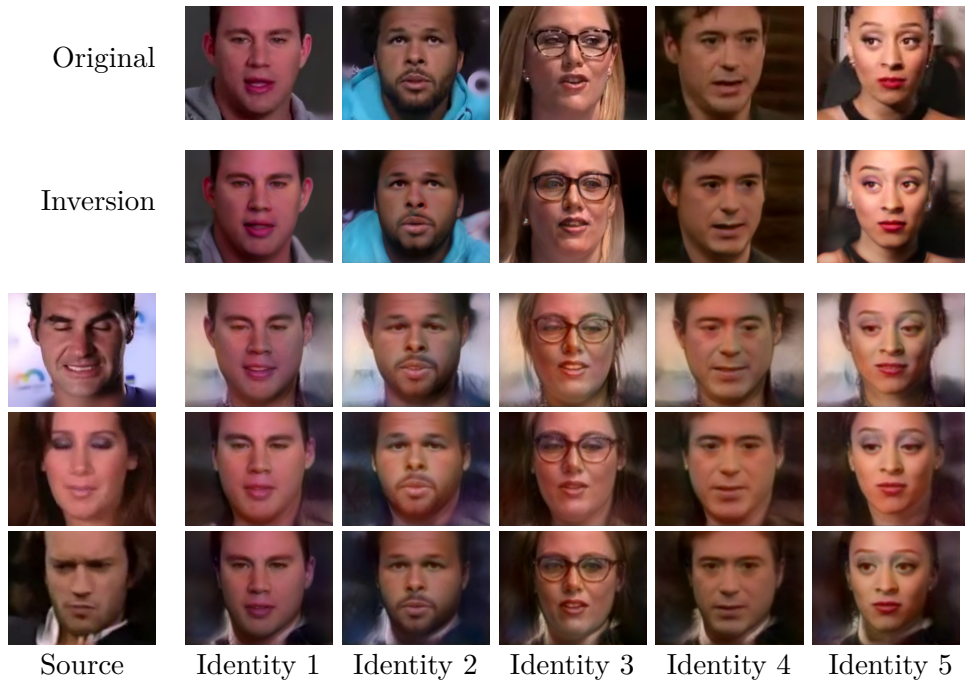

Figure 4.4: Limitations of the pose and expression transfer of our best model. The Top box depicts the identity input image along with their inversion. The first two rows show that our model has trouble generating faces with fully closed eyes and the last row shows the issue with transferring frown expression.

Limitations. Our approach has some limitations, which are illustrated in Figure 4.4. The model struggles to generate frowning faces or faces with fully closed eyes. This could be attributed to the pre-trained StyleGAN model, which was trained on the FFHQ dataset [30]. Since the dataset is mainly composed of high-quality pictures crawled from Flickr, typical expressions present are neutral and smiling expressions. Other expressions are underrepresented. Additionally, generating frowns is particularly challenging since our network takes only a single identity image and has no information about the person's neutral eyebrows position.

Expression transfer to synthetic faces. Figure 4.5 shows the expression and pose transfer performed to randomly generated identities via StyleGAN. We sample a random latent code $\mathbf{z}$ from the Gaussian distribution, from which the StyleGAN's mapping network produces $\mathbf{w} \in \mathcal{W}$. To obtain a valid identity latent code for our network, we first generate an image using StyleGAN with $\mathbf{w}$ and then invert it using ReStyle. This is due to the fact that ReStyle encodes the images into a specific subspace of StyleGAN's latent space, and the model is trained to operate in this subspace. Interestingly, using the $\mathbf{w}$ directly results in severe artifacts.

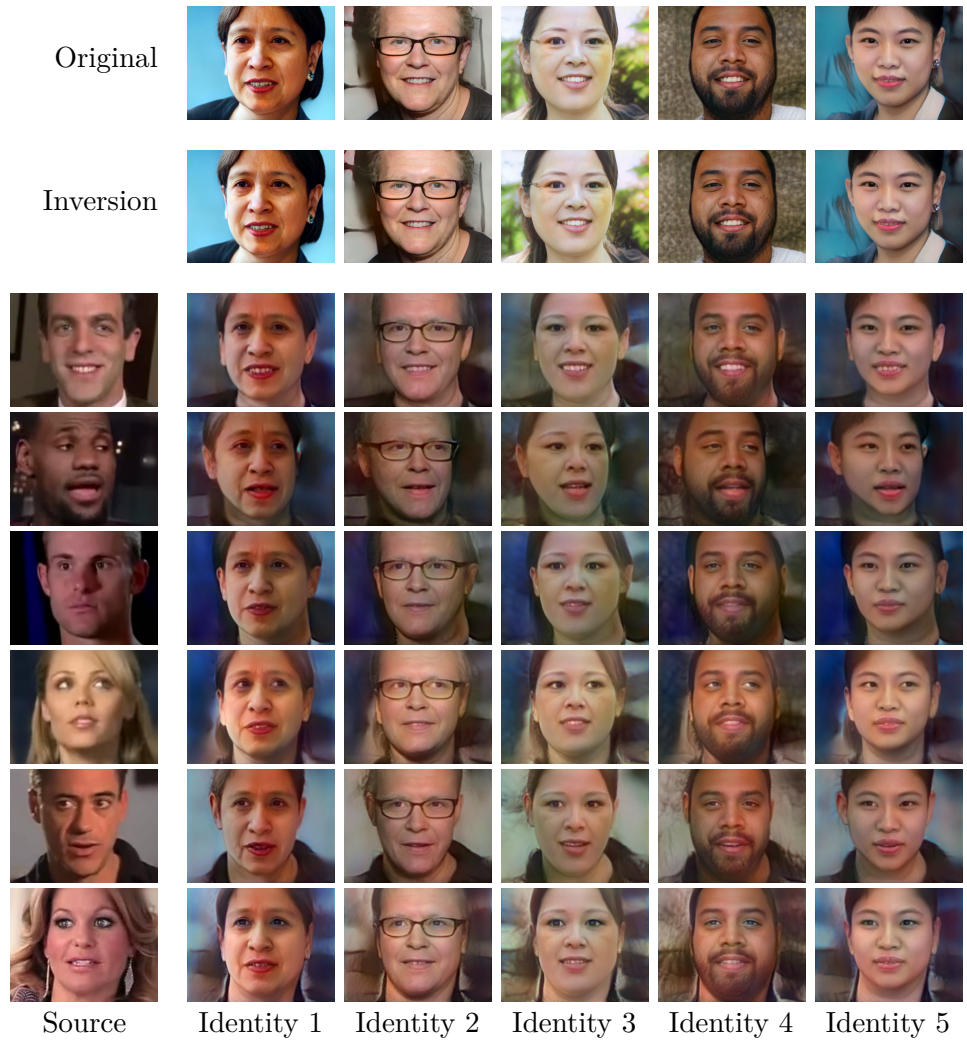

Figure 4.5: Pose and expression transfer with randomly generated identities via StyleGAN. The Top box depicts the identity input image along with their inversion.

4.4 Quantitative evaluation

We evaluate the proposed method on pose and expression transfer fidelity as well as on identity preservation. We then compare the results to the baseline methods and other variants of our method. The evaluation is done on a test split of the VoxCeleb2 dataset [48] which contains 120 different identities. The test set is preprocessed in the same way as the training one. Our evaluation focuses on a cross-reenactment scenario, where the source and target images are from different identities.

For pose transfer evaluation, we use a pre-trained CNN called HopeNet [51] which demonstrates excellent results in head pose estimation. The network outputs the predicted yaw, pitch, and roll, however, we consider only yaw and pitch as all of the preprocessed and generated images have the same roll. We calculate the mean absolute error of yaw and pitch between the generated images and their corresponding driving images. For the evaluation of identity preservation we use the already mentioned ArcFace [46]. To the best of our knowledge, there is no straightforward method for measuring expression transfer fidelity. Therefore, we utilize two different approaches – Facial Landmarks and Facial Action Units – for this task.

4.4.1 Facial Landmarks

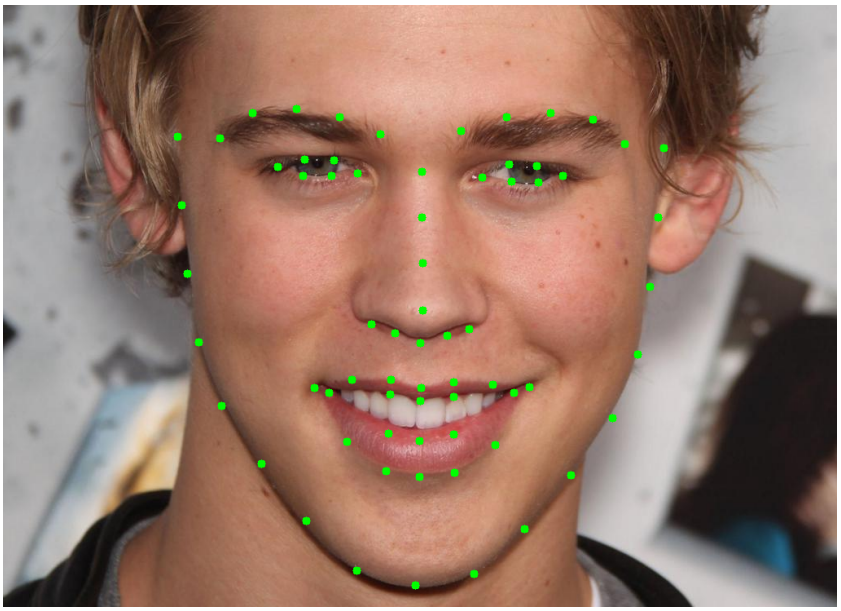

Figure 4.6: The 68 annotated landmarks on a human face.

Facial landmarks are a set of specific points on a human face that are used to locate and distinguish distinct parts of the face, such as the eyes, nose, mouth, jaw, and eyebrows. For the prediction of facial landmarks, we utilize the dlib library [49] which predicts 68 landmarks on a human face as shown in figure 4.6.

To measure the expression transfer, we utilize facial landmarks for calculating the aspect ratios of certain facial features [52]. Specifically, we measure the movement of the eyebrows by calculating the aspect ratio between both eyebrows and the eyes. To calculate the movement of the mouth and eyes, we also calculate their respective aspect ratios.

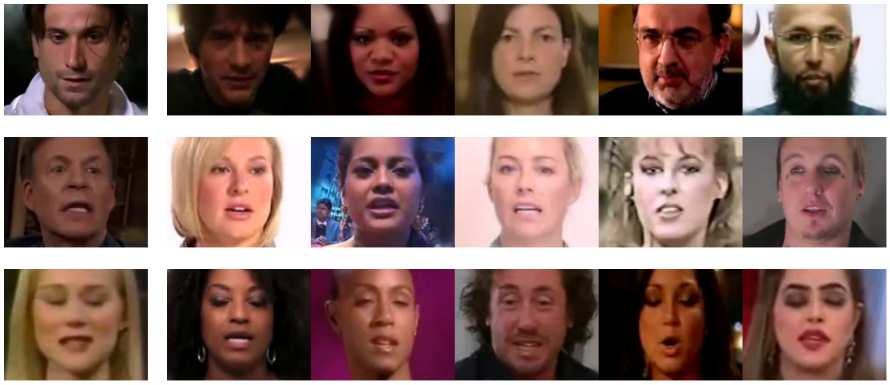

Figure 4.7: Facial landmark retrieval task where each row corresponds to a retrieval query. Specifically, the aspect ratios of the eyes, eyebrows, and mouth of the target face in the first column are queried against the training dataset, and the 5 best-matching results are retrieved.

Two facial images with the same expression should have similar aspect ratios of the facial landmarks. Figure 4.7 shows a retrieval task performed on our training dataset. We Take the 5 calculated aspect ratios of the queried face and look for the best matching results. We calculate the best matching results by minimizing the mean squared error of the aspect ratios between the queried and retrieved image. The results show that this simple approach works quite well as the query returns people with similar expressions regardless of pose.

However, this metric does not work perfectly. It does not track eye movement at all and does not measure asymmetric expressions very well (e.g. mouth movement only on one side). Another issue is that people differ in their facial structure. That is why instead of evaluating the expression transfer between single images, we calculate the correlation of aspect ratios between videos. For each frame from a source video, we generate a corresponding output image with a specified identity from a target image. Then, we calculate the correlation of each aspect ratio between the source and generated images. Finally, we average these correlations across all aspect ratios to obtain an evaluation of the expression transfer using facial landmarks.

4.4.2 Facial Action Units

Facial Action Coding System [53] represents the human face by a set of facial muscle movements called AU (Action Units). Compared with the emotion-based categorical facial expression model, AUs describe human facial expressions more comprehensively and objectively [54]. Figure 4.8 shows some of

the Facial Action Units.

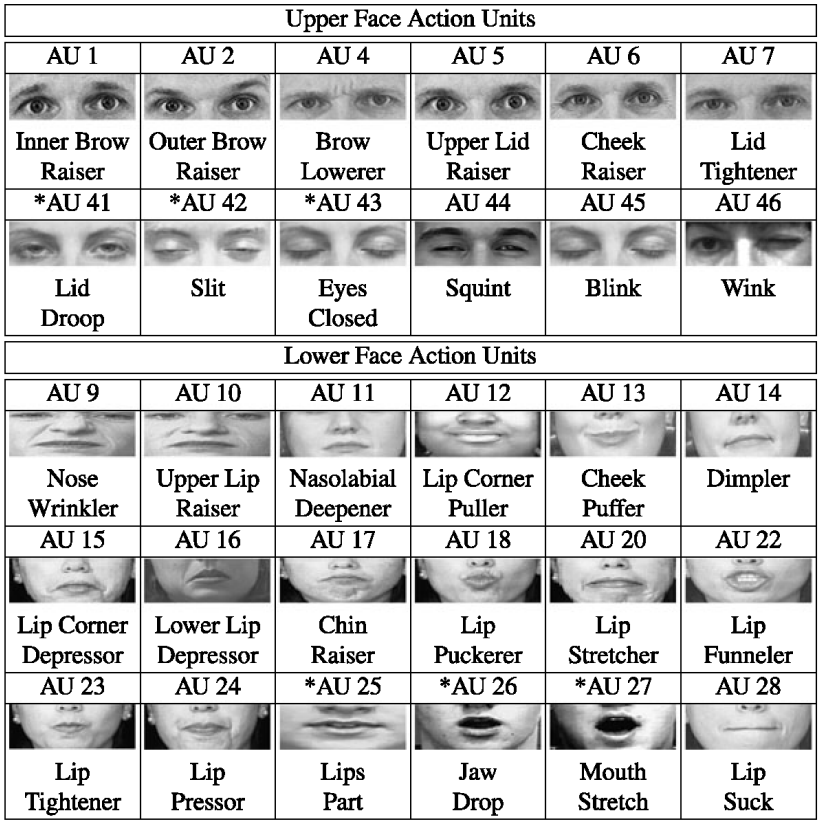

Figure 4.8: Examples of the Facial Action Units [55].

Facial AU recognition is a multi-label classification problem as multiple AUs can be activated simultaneously. For this task, we employ the graph-based neural network OpenGraphAU [54] which achieves state-of-the-art results. A recent version of their model has been trained on a dataset of 2 million in-the-wild images and can predict 28 AUs. Using this model, we can determine the probability score of individual AUs activated in a facial expression.

Two facial images with the same expression should have the same AUs activated regardless of the pose or identity of the person. Figure 4.9 shows a retrieval task performed on our training dataset. We Take the AUs of the queried face and look for the best matching results. We calculate the best matching results by minimizing the MSE of AUs between the queried and retrieved image. Unfortunately, the results show that the AUs are not as effective as Facial Landmarks for our in-the-wild images.

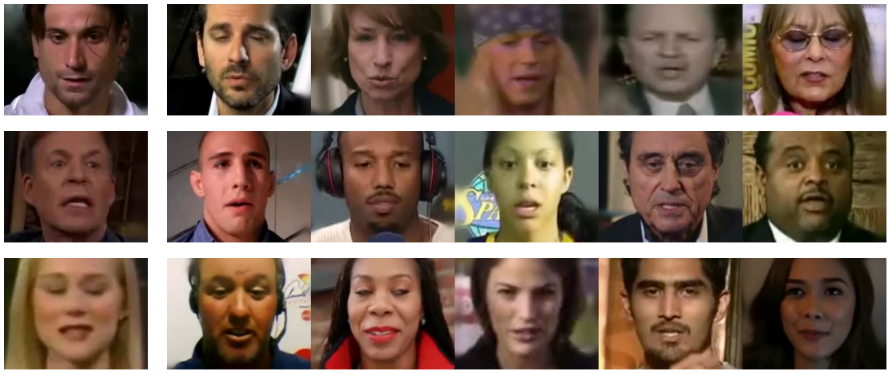

Figure 4.9: Facial Action Unit retrieval task where each row corresponds to a retrieval query. Specifically, the AUs of the target face in the first column are queried against the training dataset, and the 5 best-matching results are retrieved.

4.4.3 Overall evaluation

Table 4.1: Quantitative comparison of the baseline method and variants of our method. The first two rows show the baseline method results, first with pSp inversion configuration and second with e4e. The last three rows depict the results of our method, the first with the generator weight optimization, the second with utilizing the CosFace loss, and the last shows the best parameter model. Symbol ↑ indicates that larger is better and ↓ that smaller is better.

Method	Pose(MAE)↓	FL(CORR)↑	FAU(CORR)↑	ID(CSIM)↑
Base pSp	8.491	**0.656**	**0.210**	0.671
Base e4e	8.720	0.621	0.113	0.563
Ours Gen	8.325	0.556	0.002	0.760
Ours Cos	7.968	0.528	0.082	0.762
Ours	**7.673**	0.620	0.142	**0.801**

Table 4.1 shows the quantitative comparison of the baseline method and variants of our method on the VoxCeleb test set. Although the baseline method does transfer expression slightly better, it struggles with preserving the identity of the generated person. It also transfers the pose worse than our approach. The identity preservation is measured by the cosine similarity of ArcFace [46] embeddings. Our best model achieves a cosine similarity of 0.8, which is very good considering that the cosine similarity between the original and inverted images via Restyle with pSp configuration is 0.83. Therefore, our method achieves identity preservation close to the maximum possible with Restyle.

As mentioned before, the Facial Action Units do not seem to work very well with our images, which is not only apparent from the qualitative retrieval task but also from the calculated correlation, which is in all methods close to zero. That is why we take the correlation of facial landmarks' aspect ratios as a better indicator of expression transfer fidelity. For the pose measurement, an absolute error of yaw and pitch in degrees is calculated as the roll is always the same because of the alignment process.

Our method performs worse with the added CosFace loss function. While the loss function improves image illumination, similar to the Megaportraits paper [20], it significantly slowed down training and hindered the expression as well as eye movement transfer. The method with added generator weights also produces overall inferior output compared to the one without such optimization. The generated images suffer from more artifacts while also having a less realistic color scheme. This is probably a consequence of overfitting.

Computational demands. The speed of inference is a very important criterion. Our method needs to invert the identity image via ReStyle, which takes approximately half a second on a modern GPU. It can then generate up to 20 images per second with that identity, given all the images are already aligned. On the other hand, the baseline method requires the inversion of all the images from the source video and target video but then can generate up to 50 images per second. Given two short 5-second videos with 24 frames per second, which are typical for the VoxCeleb2 dataset, our method generates each frame from one video with the identity from the other in less than 6 seconds, whereas the baseline method would require a little over 2 minutes.

4.5 Discussion

We tried using Facial Action Units to represent expressions. We utilized a Multilayer perceptron as the motion encoder E_m which would take Facial Action Units and pose estimate of the source image and encode this to $\mathcal{W}^+$. However, this performed poorly, as the generated image would only have the pose changed slightly, seemingly ignoring the entirety of Facial Action Units. The reason is the inaccurate prediction of FAUs on our data as discussed in 4.4.2.

Conclusion

In the thesis, we presented a method for transferring pose and expression of a source face image to a target face image, while preserving the identity of the target face. The proposed method is self-supervised, and it does not require labeled data. Additionally, it fully relies on neural rendering in a one-shot setting without using a 3D graphics model of the human face.

We reviewed the existing methods and proposed a new one that is based on the StyleGAN generator. We extensively evaluated our method on pose and expression transfer fidelity as well as on identity preservation. We compare our method to the baseline that utilizes the arithmetic property of StyleGANs latent space.

Our network can transfer pose, expression, and even eye movement in close to real-time while maintaining the person's identity, even under challenging conditions such as varied ethnicity or gender. The network can handle people wearing eyeglasses as well. However, our approach has limitations in transferring certain expressions faithfully (e.g. closed eyes). Additionally, it does not achieve a perfect HD photo-realism and cannot match the visual quality of the current state-of-the-art methods that do not utilize StyleGAN.

The performance could surely be improved if we were able to get access to a better-fitting dataset. We trained on low-quality compressed videos of 224×224 pixels, where faces occupied even a smaller area of the image, while StyleGAN2 produces 1024×1024 images. Another possible improvement could be achieved by pretraining our own generator because StyleGAN2 does not offer the desired expression variability as it has been trained on images from Flickr where closed eyes and strong expressions are heavily underrepresented.

Bibliography

[1] Karras, T.; Laine, S.; et al. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2020, pp. 8110–8119, doi: 10.1109/cvpr42600.2020.00813.

[2] Goodfellow, I.; Pouget-Abadie, J.; et al. Generative Adversarial Networks. In *Advances in Neural Information Processing Systems,* [Online], volume 27, Curran Associates, Inc., 2014, pp. 2672–2680, doi: 10.1145/3422622.

[3] Murphy, K. P. *Machine learning: a probabilistic perspective*. Cambridge: MIT Press, 2012, ISBN 978-0-262-01802-9. Available from: `https://probml.github.io/pml-book/book0.html`

[4] Goodfellow, I.; Bengio, Y.; et al. *Deep Learning*. MIT Press, 2016, ISBN 978-0-262-33737-3. Available from: `http://www.deeplearningbook.org`

[5] Cootes, T. F.; Edwards, G. J.; et al. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence,* [Online], volume 23, no. 6, 2001: pp. 681–685, doi:10.1109/34.927467.

[6] Blanz, V.; Vetter, T. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* [Online], 1999, pp. 187–194, doi:10.1145/311535.311556.

[7] Vlasic, D.; Brand, M.; et al. Face Transfer with Multilinear Models. *ACM Trans. Graph.,* [Online], volume 24, no. 3, 2005: p. 426–433, doi:10.1145/1073204.1073209.

[8] Thies, J.; Zollhöfer, M.; et al. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.,* [Online], volume 34, no. 6, 2015: pp. 183–1, doi:10.1145/2816795.2818056.

[9] Thies, J.; Zollhofer, M.; et al. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* [Online], 2016, pp. 2387–2395, doi:10.1109/cvpr.2016.262.

[10] Zollhöfer, M.; Thies, J.; et al. State of the art on monocular 3D face reconstruction, tracking, and applications. In *Computer graphics forum,* [Online], volume 37, Wiley Online Library, 2018, pp. 523–550.

[11] Kim, H.; Garrido, P.; et al. Deep video portraits. *ACM Transactions on Graphics (TOG),* [Online], volume 37, no. 4, 2018: pp. 1–14, doi: 10.1145/3197517.3201283.

[12] Lombardi, S.; Saragih, J.; et al. Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG),* [Online], volume 37, no. 4, 2018: pp. 1–13, doi:10.1145/3197517.3201401.

[13] Doukas, M. C.; Zafeiriou, S.; et al. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* [Online], 2021, pp. 14398–14407, doi: 10.1109/iccv48922.2021.01413.

[14] Ding, H.; Sricharan, K.; et al. Exprgan: Facial expression editing with controllable expression intensity. *Proceedings of the AAAI conference on artificial intelligence,* [Online], volume 32, no. 7, 2018, doi: 10.1609/aaai.v32i1.12277.

[15] Worrall, D. E.; Garbin, S. J.; et al. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision,* [Online], 2017, pp. 5726–5735, doi: 10.1109/iccv.2017.611.

[16] Qiao, F.; Yao, N.; et al. Geometry-contrastive GAN for facial expression transfer. *arXiv preprint arXiv:1802.01822,* [Online], 2018, doi:10.48550/arXiv.1802.01822.

[17] Wiles, O.; Koepke, A.; et al. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV),* [Online], 2018, pp. 670–686, doi:10.1007/978-3-030-01261-8_41.

[18] Zakharov, E.; Shysheya, A.; et al. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF international conference on computer vision,* [Online], 2019, pp. 9459–9468, doi:10.1109/iccv.2019.00955.

[19] Wang, T.-C.; Mallya, A.; et al. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2021, pp. 10039–10049, doi:cvpr46437.2021.00991.

[20] Drobyshev, N.; Chelishev, J.; et al. Megaportraits: One-shot megapixel neural head avatars. *arXiv preprint arXiv:2207.07621,* [Online], 2022, doi:10.48550/arXiv.2207.07621.

[21] Nguyen, T. T.; Nguyen, Q. V. H.; et al. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding,* [Online], volume 223, 2022: p. 103525, doi:10.2139/ssrn.4030341.

[22] Zhu, J.-Y.; Park, T.; et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision,* [Online], 2017, pp. 2223–2232, doi:10.1109/iccv.2017.244.

[23] Härkönen, E.; Hertzmann, A.; et al. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems,* [Online], volume 33, 2020: pp. 9841–9850, doi:10.48550/arXiv.2004.02546.

[24] Shen, Y.; Gu, J.; et al. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2020, pp. 9243–9252, doi:10.1109/cvpr42600.2020.00926.

[25] Abdal, R.; Zhu, P.; et al. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG),* [Online], volume 40, no. 3, 2021: pp. 1–21, doi:10.1145/3447648.

[26] Tewari, A.; Elgharib, M.; et al. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* [Online], 2020, pp. 6142–6151, doi:10.1109/cvpr42600.2020.00618.

[27] Radford, A.; Metz, L.; et al. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434,* [Online], 2015, doi:10.48550/arXiv.1511.06434.

[28] Arjovsky, M.; Chintala, S.; et al. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning,* [Online], PMLR, 2017, pp. 214–223, doi:10.48550/arXiv:1701.07875.1701.07875.

[29] Karras, T.; Aila, T.; et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196,* [Online], 2017, doi:10.48550/arXiv.1710.10196.

[30] Karras, T.; Laine, S.; et al. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2019, pp. 4401–4410, doi:10.1109/cvpr42600.2020.00813.

[31] Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision,* [Online], 2017, pp. 1501–1510, doi:10.1109/iccv.2017.167.

[32] Abdal, R.; Qin, Y.; et al. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2020, pp. 8296–8305, doi:10.1109/iccv.2019.00453.

[33] Petrželková, N. Face Image Editing in Latent Space of Generative Adversarial Networks. Prague, 2021, Bachelor thesis. CTU in Prague, Faculty of Electrical Engineering, Department of Cybernetics.

[34] Creswell, A.; Bharath, A. A. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems,* [Online], volume 30, no. 7, 2018: pp. 1967–1974, doi:10.1109/tnnls.2018.2875194.

[35] Zhu, J.-Y.; Krähenbühl, P.; et al. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14,* [Online], Springer, 2016, pp. 597–613, doi:10.1007/978-3-319-46454-1_36.

[36] Šubrtová, A.; Futschik, D.; et al. ChunkyGAN: Real Image Inversion via Segments. In *Proceedings of European Conference on Computer Vision,* [Online], 2022, pp. 189–204.

[37] Abdal, R.; Qin, Y.; et al. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* [Online], 2019, pp. 4432–4441, doi:10.1109/iccv.2019.00453.

[38] Roich, D.; Mokady, R.; et al. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG),* [Online], volume 42, no. 1, 2022: pp. 1–13, doi:10.1145/3544777.

[39] Zhu, P.; Abdal, R.; et al. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036,* [Online], 2020, doi: 10.48550/arXiv.2012.09036.

[40] Richardson, E.; Alaluf, Y.; et al. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2021, pp. 2287–2296, doi:10.1109/cvpr46437.2021.00232.

[41] Alaluf, Y.; Patashnik, O.; et al. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* [Online], 2021, pp. 6711–6720, doi:10.1109/iccv48922.2021.00664.

[42] Tov, O.; Alaluf, Y.; et al. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG),* [Online], volume 40, no. 4, 2021: pp. 1–14, doi:10.1145/3450626.3459838.

[43] Tzaban, R.; Mokady, R.; et al. Stitch it in time: Gan-based facial editing of real videos. In *SIGGRAPH Asia 2022 Conference Papers,* [Online], 2022, pp. 1–9, doi:10.1145/3550469.3555382.

[44] Subrtova, A.; Cech, J.; et al. Hairstyle Transfer between Face Images. *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021),* [Online], 2021: pp. 1–8, doi:10.1109/FG52635.2021.9667038.

[45] Zhang, R.; Isola, P.; et al. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* [Online], 2018, pp. 586–595, doi: 10.1109/cvpr.2018.00068.

[46] Deng, J.; Guo, J.; et al. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* [Online], 2019, pp. 4690–4699, doi: 10.1109/cvpr.2019.00482.

[47] Wang, H.; Wang, Y.; et al. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* [Online], 2018, pp. 5265–5274, doi: 10.1109/cvpr.2018.00552.

[48] Chung, J. S.; Nagrani, A.; et al. VoxCeleb2: Deep Speaker Recognition. In *Proc. Interspeech 2018,* [Online], 2018, pp. 1086–1090, doi:10.21437/Interspeech.2018-1929.

[49] King, D. E. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research,* [Online], volume 10, 2009: pp. 1755–1758, doi: 10.5555/1577069.1755843.

[50] Wright, L.; Demeure, N. Ranger21: a synergistic deep learning optimizer. *CoRR,* [Online], volume abs/2106.13731, 2021, doi:arXiv:2106.13731, 2106.13731.

[51] Ruiz, N.; Chong, E.; et al. Fine-Grained Head Pose Estimation Without Keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* [Online], 2018, pp. 2074–2083, doi: 10.1109/cvprw.2018.00281.

[52] Soukupova, T.; Cech, J. Eye blink detection using facial landmarks. In *21st computer vision winter workshop, Rimske Toplice, Slovenia,* [Online], 2016, p. 2.

[53] Ekman, P.; Friesen, W. V. Facial action coding system. *Environmental Psychology & Nonverbal Behavior,* [Online], 1978, doi:10.1037/t27734-000.

[54] Luo, C.; Song, S.; et al. Learning multi-dimensional edge feature-based au relation graph for facial action unit recognition. *arXiv preprint arXiv:2205.01782,* [Online], 2022, doi:10.24963/ijcai.2022/173.

[55] Zhi, R.; Liu, M.; et al. A comprehensive survey on automatic facial action unit analysis. *The Visual Computer,* [Online], volume 36, 2020: pp. 1067–1093, doi:10.1007/s00371-019-01707-5.

Appendix A

Acronyms

3DMM 3D Morphable Model

AAM Active Appearance Model

AdaIN Adaptive Instance Normalization

CNN Convolutional Neural Network

DCGAN Deep Convolutional Generative Adversarial Network

FAU Facial Action Units

FL Facial Landmarks

GAN Generative Adversarial Network

MSE Mean Squared Error

PCA Principal Component Analysis

SVM Support Vector Machine

Appendix B

Contents of enclosed archive

- `README.md` ..... archive contents description
- `src` ..... source codes
  - `README.md` ..... instructions for local inference
- `videos` ..... generated videos
- `text` ..... thesis text
  - `BP_Jahoda_Petr_2023.pdf` ..... thesis text in PDF format
  - `src` ..... LaTeX source codes of the thesis