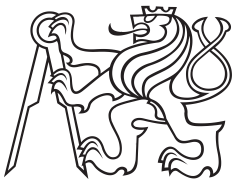
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Hairstyle Transfer between Portrait Images

Bc. Adéla Šubrtová

Supervisor: Ing. Jan Čech, Ph.D.
Field of study: Open Informatics
Subfield: Computer Vision and Image Processing
May 2021

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Šubrtová Adéla** Personal ID number: **457114**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Computer Vision and Image Processing**

II. Master's thesis details

Master's thesis title in English:

Hairstyle Transfer between Portrait Images

Master's thesis title in Czech:

Přenos účesu mezi portréty

Guidelines:

Propose a method that takes two portrait images as an input (source and target) and produces an image where the hairstyle is transferred from one image to the other, i.e. the output image will be a realistic composition consisting of the inner face of the target image while the hairs are of the source image. The input images are not assumed to be geometrically aligned. Focus on convolutional neural networks with an encoder-decoder architecture and consider using a pre-trained photo-realistic Generative Adversarial Network, e.g. [1].

Bibliography / sources:

[1] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. Proc. CVPR, 2020.
[2] Elad Richardson, Yuval Alaluf, Or Patashnik,Yotam Nitzan, Yaniv Azar, Stav Shapiro, Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. Proc. ICLR, 2021. (In review)
[3] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, Nenghai Yu. MichiGAN: Multi-Input-Conditioned Hair Image Generation for Portrait Editing. ACM Transactions on Graphics (TOG), vol. 39, num. 4, 2020.

Name and workplace of master's thesis supervisor:

Ing. Jan Čech, Ph.D., Visual Recognition Group, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **07.01.2021** Deadline for master's thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Ing. Jan Čech, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would like to express sincere gratitude to my supervisor Ing. Jan Čech, Ph.D. for his leadership and encouragement in the last four years. Special thanks to Ing. Vojtěch Franc, Ph.D. for his valuable advice and for his patience. I really learned a lot from you both.

Many thanks to the people who participated in the user study. Lastly, I thank my family and friends for the unwavering support in the recent years.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 21, 2021

. . . . . .
signature

Abstract

This thesis proposes a compact solution for high-fidelity hairstyle transfer between portrait images. Given a hair image and a face image, our network produces an output image having the input hair and face seamlessly merged. The architecture consists of two encoders and a tiny mapping network that map the two inputs into the latent space of the pretrained StyleGAN2, which generates a high-quality image. The method needs neither annotated data nor an external dataset; the whole pipeline is trained using only synthetically generated images by the StyleGAN2.

We demonstrate additional applications of the proposed framework, e.g., hairstyle manipulation and hair generation for 3D morphable model renderings. The extensive evaluation shows that our network is robust to various challenging conditions, where a head pose, face size, gender, ethnicity, and illumination differ between the inputs. The hairstyle transfer fidelity is assessed by a user study and using a trained hair similarity metric.

Keywords: hair synthesis, hair transfer, hair manipulation, StyleGAN2, autoencoder

Supervisor: Ing. Jan Čech, Ph.D.

Abstrakt

Tato diplomová práce navrhuje ucelené řešení problému přenosu účesu mezi portréty. Vstupem jsou dva portréty různých lidí, snímek vlasů a snímek obličeje. Metoda vyprodukuje přirozeně vypadající portrét, který má účes ze snímku vlasů a identitu ze snímku obličeje. Navrhovaná architektura se skládá ze dvou enkodérů a menší spojovací sítě, které mapují oba vstupní snímky do latentního prostoru sítě StyleGAN2. Ta následně vygeneruje výsledný obrázek ve vysoké kvalitě. Metoda nevyžaduje anotace ani externí dataset – proces trénování je proveden na syntetických datech generovaných ze sítě StyleGAN2.

Metoda má několik dalších využití, jako je např. manipulace účesů a generování vlasů pro 3D morphable model. Z rozsáhlého vyhodnocení je zřejmé, že náš přístup je robustní vůči náročným podmínkám, a to včetně případů, kdy jsou vstupy různě natočeny a nasvíceny nebo se liší velikostí obličeje, etnicitou či pohlavím. Kvalita přenesených účesů je vyhodnocena pomocí naučené metriky pro podobnost vlasů a také pomocí uživatelské studie.

Klíčová slova: syntéza vlasů, přenos účesu, manipulace vlasů, StyleGAN2, autoenkodér

Překlad názvu: Přenos účesu mezi portréty

Contents

Figures

Tables

Chapter 1

Introduction

Hairstyles have been evolving throughout history. The first mention of hair styling dates as early as 25,000 years ago [1]. Through time, people styled, dyed, braided, and curled their hair or even wore wigs for fashion purposes. Nowadays, more so than ever, people express their individuality through hairstyles. Humans perceive hair as a part of one's identity; a person with eccentric hair is a lot more memorable than the one with a regular haircut. Since hair is such a critical element of one's visual identity, a good haircut can improve physical appearance and self-confidence. Regarding the fact, it is hard to envision how one would look with a different hairstyle. It can be of great value to generate an image of oneself with a different hairstyle before changing it significantly at the hairdresser.

However, hair is intrinsically complex and variable, and it is a very challenging problem to generate a facial image with realistic-looking hair. In 1999, Blanz and Vetter introduced a 3D morphable model for face synthesis [2]. Nevertheless, the model can only generate the inner part of the face. With the introduction of Generative Adversarial Networks (GANs) [3], the generation of realistic faces became more accessible. The current state-of-the-art method for realistic image synthesis is StyleGAN2 [4] with output resolution of 1024×1024 pixels. However, realistic hair generation and manipulation are still not sufficiently explored.

The GANs produce photorealistic images; but, they are known for their intricate training, which requires a large amount of data, not to mention the need for annotated data in most cases. We propose a solution to overcome both of those shortcomings.

This thesis aims to implement an end-to-end framework for hair generation with other applications such as hairstyle transfer between two images and hair generation for a 3D morphable model. Given two images, a face image and a hair image, our framework generates a high-fidelity output image with the face and hair merged seamlessly. To bypass the precarious training of a GAN network, we employ *fixed* pretrained StyleGAN2 network [4]. Furthermore, the training is conducted on an entirely synthetic dataset generated by the StyleGAN2, i.e., it does neither require any external dataset nor any annotations. The applications can be seen in Figure 1.1.

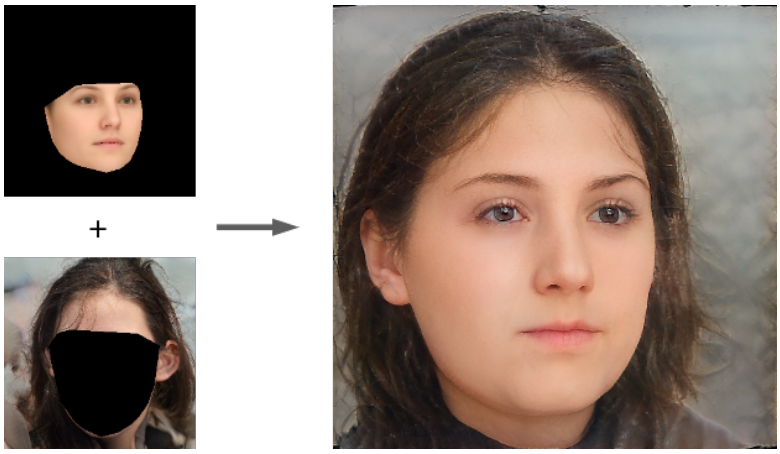
We evaluate the method both quantitatively and qualitatively. The quanti-

tative evaluation is conducted by a learned hair similarity metric and a user study. The qualitative assessment consists of several experiments, namely, hairstyle transfer, interpolation, and hair generation for 3D morphable model renderings, latent space manipulation, and comparison with other state-of-the-art methods.

The thesis presents three primary contributions. First, the proposed method does not require any annotations or an external dataset. Second, we evade the intricate training of GANs by employing the pretrained StyleGAN2 generator. Consequently, the output images attain high resolution of 1024×1024 pixels. Lastly, we train a hair similarity metric to quantitatively assess the hair transfer performance of the proposed method. Part of the thesis is submitted to a conference [5].

The rest of this thesis is structured as follows. Chapter 2 presents the related work regarding hair modeling, generation, realistic image generation, and face swapping. The technical background regarding StyleGAN2 and 3D morphable model is described in Chapter 3. The architecture, dataset, and training details are described in Chapter 4. Chapter 5 contains an extensive evaluation. Finally, the thesis is summarized in Chapter 6.

(a) : Hairstyle transfer



(b) : 3D morphable model

Figure 1.1: Applications of the proposed method. (a) High-fidelity hair transfer, (b) 3D morphable model hair generation.

Chapter 2

Related Work

A significant number of methods have been proposed for the acquisition of models capturing hair geometry and its appearance from 2D images [6, 7, 8, 9, 10, 11, 12, 13]. The models have been used for various purposes like, for example, hairstyle transfer [8, 14], animation [9], or morphing [10]. A survey of modeling techniques for hairstyling, hair simulation, and hair rendering can be found in [15]. The geometric models often capture only a coarse hair geometry and its properties that do not allow for the generation of photorealistic images.

With the advent of deep learning, there has been a paradigm shift to using neural networks as the underlying model for face synthesis. An efficient algorithm for learning deep generative models was introduced in the seminal work [3] which proposed the Generative Adversarial Networks (GANs). The GANs allow to generate a high-dimensional distribution by transforming a fixed low-dimensional distribution via a neural network. The parameters of the network are learned such that the distribution of the high-dimensional outputs matches the training data. A surge of papers applying GANs to image modeling was initiated by a fully convolutional architecture DCGAN [16]. Since then, the quality of images synthetized by GANs has been steadily growing [17, 18, 19]. The current state-of-the-art is represented by the StyleGAN2 achitecture [4] which produces high-resolution photorealistic faces that are for humans hard to distinguish from real photos.

In our work, we use the StyleGAN2 as the underlying generator of the facial images. A similar idea was previously considered by others in e.g., [20, 21, 22]. In [20], the authors propose an embedding algorithm transforming a given photograph into the latent space of StyleGAN. The algorithm starts from a randomly generated latent code, which is then optimized by a gradient method until the output of StyleGAN matches the input image in terms of perceptual [23] and L_2 loss. The authors investigate how algebraic operations on the latent codes correspond to semantic processing operations on the images. The work [21] addresses the problem of interactive image editing via manipulation of latent codes. As the backbone generator, they use pretrained progessive GAN [17]. The latent codes are computed by the same iterative algorithm as in [20]. An encoder network directly mapping the input images into latent vectors which are fed to StyleGAN was proposed in [22]. The encoder is

trained by minimizing a combination of L_2-loss, perceptual loss, and identity capturing loss that are used to evaluate a similarity between the input and the generated images. The authors show that the direct encoding works on par or better than the methods optimizing the latent vector iteratively [20, 21]. In our work, we exploit the same idea of directly mapping the image to the latent vectors and we also use a similar loss function, however, we do not require real images for training.

Deep generative networks specifically designed for hairstyle modeling were proposed in [24, 25, 26]. The work [24] introduces *Hairstyle30k* database which is composed of 30k face images, each labeled by one out of 64 different hairstyles annotated by a semiautomated process. The database is exploited to train H-GAN model which is a variant of the Conditional Variational Auto-Encoder. Using the conditioning allows the H-GAN to explicitly control the hairstyle of the generated facial images. The authors show how to use their model for hairstyle classification and hair editing in portrait images. In our work, we use a different architecture and we do not require annotated faces for training. RSGAN [25] is a generative neural network designed for face swapping and edditing of facial attributes. Similarly to our approach, they train two encoding networks, one for the inner face and one for the hair region. The latent representations of the inner face and the hair region input the generator network which outputs the synthesized face image. In contrast, our approach uses the pretrained StyleGAN2 generator, while they train all components of the architecture from scratch using annotated face images. MichiGAN [26] is an interactive hair editing system for portrait images. The system is based on a complicated architecture which implements a sequential conditioning mechanism composed of three condition modules. The MichiGAN architecture allows for an orthogonal control over four attributes including hair shape, structure, appearance, and image background.

To summarize, the main distinguishing feature of our approach compared to existing works is that our learning algorithm does not require any external set of training examples, instead it *learns solely from synthetic images* generated by the pretrained StyleGAN2. In contrast, the existing hairstyle modeling GANs, i.e. H-GAN[24], RSGAN [25] and MichiGAN [26], all require a large database of real faces, which in the case of H-GAN and RSGAN has to be annotated with the hairstyle. Collecting such training databases is difficult because it has to contain faces with a large variation of hairstyles, annotation of which is a tedious task with often inconsistent outcomes. Apart from this, our method generates photorealistic 1024×1024 images in contrast to existing architectures producing lower-resolution outputs, namely, 128×128 in case of H-GAN and RSGAN, and 512×512 in the case of MichiGAN.

A related problem to hairstyle transfer is face swapping, e.g. [27, 14, 28]. As opposed to hair transfer, face swapping preserves the pose of the hair input image. One of the older approaches is to search a large database for replacement candidates for a given image [27, 14]. This approach does not allow face swapping in arbitrary pair of images. Specifically, [27] uses a fixed offline-constructed database of candidate images. The swapping process

includes finding and ranking images within the database with matching pose, adjusting the appearance and ranking the final results. The image set is pruned during each step of the pipeline. Work [14] uses a slightly different technique. The search in an online database (an Internet image search engine) is conditioned on additional key-word query, which is used to further specify the candidate set. The plausible target photos are ranked by L_2 distance between the VGG features of the source and target images. The final target images are selected based on a sophisticated image similarity function considering difference in age, pose, and other factors. Masked-out hair and face are blended using gradient domain stitching [29].

Nirkin et al. [28] allow the user to perform face swap in arbitrary photos. The method employs 3D face shape estimation and inner-face segmentation to handle pairs with different poses and occlusions. Pixel values are transferred using the 3D shapes and the final image is blended using Poisson image editing [30].

Chapter 3

Technical Background

3.1 Generative Adversarial Networks

Generative Adversarial Network (GAN) [3] is a framework consisting of two networks; Generator and Discriminator. The Generator G aims to generate data with a distribution that matches the distribution of the training set. The generation resides in transforming a simple low-dimensional distribution into a more complex distribution of high dimension. The Discriminator D is trained to discern between the real image (from the training distribution) or the generated image by the Generator. The problem is formulated as a mini-max game between G and D, which results in a very fragile training process.

Due to the complexity of the optimization criteria, convergence is not always assured. Many papers focus on improving the stability by changing the objective, such as Wasserstein GAN [31], or balancing G and D, Boundary Equilibrium GAN [32]. Nevertheless, the generation of high-resolution output images is still hard to achieve.

In 2018, Karras et al. from NVIDIA [17] proposed a new way to train GANs. By training both G and D progressively, meaning that at first, the framework is learned to generate images with low resolution and by adding layers during training, the resolution gradually increases. This process stabilizes training and allows the networks to generate high-quality images of size 1024×1024.

Until 2019, the common image generation architecture was the classical deep convolutional GAN (DCGAN) [16]. Inspired by style transfer, Karras et al. [19] introduced the StyleGAN network.

3.2 StyleGAN

There are several differences between the traditional DCGAN and the StyleGAN. Firstly, the DCGAN feeds the sampled latent code into the first layer. The StyleGAN approach samples the latent vector and maps it into higher-dimensional intermediate latent space, obtaining several style codes. Each

of the codes is transformed by channel-wise AdaIN operation[1] [33] and fed into a designated convolution layer. The input of the first layer is a constant vector, which is optimized during training. The layout is depicted in Figure 3.1 taken from the original paper [19]. The style codes can be separated into three categories; coarse, middle, and fine styles, depending on the resolution they influence. Coarse style vectors (resolution $4^2 - 8^2$) influence higher-level features such as pose, face shape, and general hairstyle. Middle style codes (res. $16^2 - 32^2$) determine facial features such as expression. Finally, the fine style vectors (res. $64^2 - 1024^2$) define the overall color scheme.

Per-pixel noise is injected after each convolution to control the stochastic variation of the exact placement of hair or freckles and others. The network is trained progressively starting at 4×4 resolution and finishing at 1024×1024 pixels.

In the subsequent paper of same authors [4], it is discovered that the progressive training causes certain artifacts, e.g., teeth stay facing the camera in nonfrontal poses. Thus, the authors relinquish the progressive training in favor of a more elaborate architecture with skip connections between low-dimensional feature maps and the final image. Furthermore, the improved version enforces smoothness of the latent space by path length regularization, making it easier to interpolate in the latent space and to invert the network (i.e., to find the corresponding latent code for a given image).

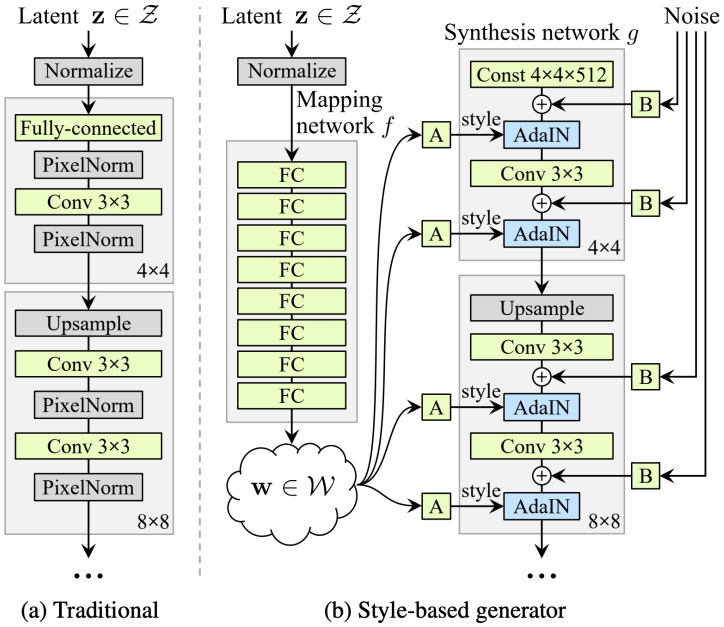


Figure 3.1: Comparison of the style-based architecture (b) to the classic approach (a). The style codes are transformed by AdaIn - denoted as "A" and added after each convolution. Noise is injected to generate stochastic variation. "B" denotes learned per-channel scaling. The figure is taken from the original paper [19].

[1] Adaptive instance normalization

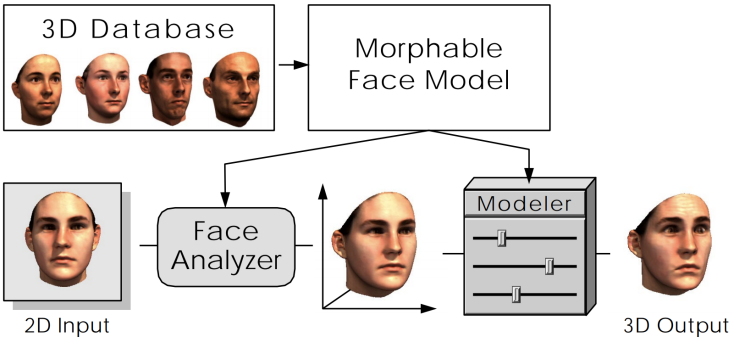


Figure 3.2: The 3D morphable model scheme. The 3D face model can be generated by random parameters or estimated from a given 2D input image. The figure is taken from the original publication [2].

3.3 3D Morphable Model

3D Morphable model (3DMM) [2] is a parametric model for face modeling. Shape, texture, and illumination are managed separately.

The model is constructed from a dataset containing many facial scans of different people. Each scan consists of a 3D mesh and texture. The mesh vertices are registered i.e, the vertices have the same semantic label for all subjects in the dataset.

Linear Principal Component Analysis (PCA) is employed to build a statistical model of the dataset. The model is encoded into two parts: a mean face model, and two covariance matrices for shape and texture. The PCA is used to compress the model and to remove the correlation between the samples in the dataset by transforming the basis to an orthogonal coordinate system. The orthogonal basis vectors represent the directions of the highest variance in the data in decreasing order (i.e., the first component has the highest variance). While the most prominent directions can be assigned semantic value such as gender, the directions with the lowest variance can be neglected to reduce the dimensionality of the model. The axes of the orthogonal system are the eigenvectors of the covariance matrix.

A new face is modelled as a linear combination of the basis vectors added to the mean face, independently for shape and texture. 2D images of the face views are rendered using a standard rendering pipeline.

3DMM parameters can be found for a single face image. The 3DMM obtains the 3D shape of the face, texture, pose, and illumination by optimizing the image difference between the model rendering and the input image over the 3DMM parameters. The 3DMM scheme is depicted in Figure 3.2.

Since its introduction, the morphable model has become increasingly detailed. However, the morphable model still cannot generate realistic-looking hair. Mainly due to the fact that the identities need to be in full correspondence.

Chapter 4

Proposed Method

Our framework takes two input images, a facial image and a hair image, and encodes them into embeddings in latent spaces. Both embeddings are then mapped into an embedding of the latent space of the pretrained StyleGAN2, which generates the high-resolution output image.

4.1 Architecture

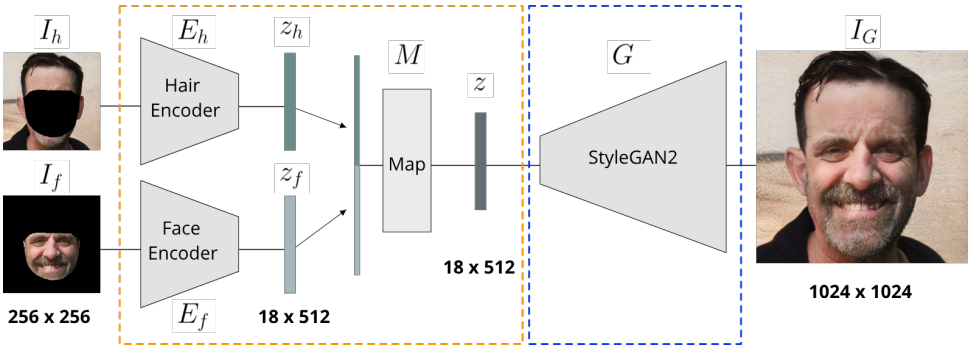


Figure 4.1: The architecture of the proposed hairstyle transfer framework. All the networks in the orange rectangle are updated during training. The weights of the network in the blue rectangle stay fixed.

The proposed architecture is depicted in the Figure 4.1. The framework consists of two encoding networks, hair encoder E_h and face encoder E_f, a fully connected mapping network M, and a generator network G.

Given an input image I of size 256 $\times$ 256 pixels, we create the network inputs as follows. Firstly, in I we find 2D facial landmarks using [34] and create their convex hull, getting a binary mask. The facial input I_f has the hair masked out, and inversely, the hair input I_h has the inner face masked out. Formally,

$$I = I_f + I_h.$$

Both inputs are fed into their respective encoders (i.e., I_f into E_f, and I_h into E_h) obtaining two embeddings with dimensions 18$\times$512 each; z_f for the face, and z_h for the hair. Subsequently, the embeddings are concatenated

and projected by the mapping network M into z with dimensions 18×512. Finally, the network G generates the output image I_G from z.

Formally, the image generation is written as

$$I_G = G\Big(M\Big(E_h(I_h) \oplus E_f(I_f)\Big)\Big),$$

where the symbol $\oplus$ denotes concatenation.

The architecture used for E_f and E_h is ResNet-IR SE 50 adopted from the repository accompanying the paper [22]. However, the paper uses a single encoder with a more elaborate architecture based on feature pyramid networks [35]. Mapping network M consists of one fully connected layer and a LeakyReLU activation layer with a negative slope of 0.2. For generator G, we employ StyleGAN2 pretrained on FFHQ dataset [4] with high output resolution of 1024×1024.

4.2 Training

During training, the weights of G stay *fixed* and only the weights of E_f, E_h, and M are updated. The network is trained to auto-encode the input image, i.e., to generate image I_G that matches the input I. Since the input I has size 256×256 pixels and the output image I_G has 1024×1024, we subsampled I_G before computing the loss function. Let us denote the subsampled generated image $I_{G_{256}}$.

$$I_{G_{256}} = I_G \downarrow 256,$$

where $\downarrow$ represents subsampling to 256×256.

Following [22], the loss function consists of three components; pixel-wise, perceptual, and identity loss.

Pixel-wise loss. Pixel-wise loss $\mathcal{L}_{L_2}$ is simply:

$$\mathcal{L}_{L_2} = ||I - I_{G_{256}}||_2$$

Perceptual loss. In [23], the authors discovered that the deep features of neural networks pretrained on image classification tasks such as VGG or AlexNet correlate well with human perception of image similarity. The perceptual loss, denoted as $\mathcal{L}_{LPIPS}$, is computed as follows:

$$\mathcal{L}_{LPIPS} = ||F(I) - F(I_{G_{256}})||_2,$$

where F, in our case, represents the deep features of AlexNet.

Identity loss. When looking at a portrait image, most of the attention is focused on the facial area, so preserving the identity is essential. It seems that without enforcing the identity loss explicitly, the network fails to generate the necessary detail in facial features. The identity loss $\mathcal{L}_{ID}$ is realized using the ArcFace net [36] trained for face recognition task. The loss is calculated as

$$\mathcal{L}_{ID} = 1 - \langle A(\tilde{I}), A(\tilde{I}_{G_{256}})\rangle,$$

where A is the ArcFace network, $\tilde{I}$ and $\tilde{I}_{G_{256}}$ are cropped-out faces (of I and $I_{G_{256}}$ respectively) subsampled to 112×112 to fit the ArcFace input size.

The complete loss function is:

$$\mathcal{L} = \lambda_{LPIPS}\mathcal{L}_{LPIPS} + \lambda_{ID}\mathcal{L}_{ID} + \lambda_{L_2}\mathcal{L}_{L_2}.$$

The λ parameters were set following [22] for StyleGAN2 inversion task. Specifically, $\lambda_{LPIPS} = 0.8$, $\lambda_{ID} = 0.1$, and $\lambda_{L_2} = 1$.

We employed the Ranger optimizer from [22] with learning rate 0.0001 and batch size only 2 due to memory limitations. We initialized both encoders with the weights of the ResNet pretrained for face recognition [36].

4.3 Dataset

We randomly generated 70,000 images using StyleGAN2 network. We first independently sampled a 512-dimensional vector x from $\mathcal{N}(0, 1)$. Subsequently, we mapped x into the intermediate latent space of the StyleGAN2, $\mathcal{W}$, of dimension 18×512 using the mapping part of the StyleGAN2. Each sample was then manipulated in $\mathcal{W}$ to generate a 5-step sequence of yaw angles within the interval of $[-25°, 25°]$. An example of such a sequence is depicted in Fig. 4.2. In summary, the dataset contained 350,000 images.

Yaw manipulation. To manipulate the pose, we followed a standard approach, e.g., [37]. Specifically, we learned a linear classifier to distinguish between latent codes of faces with positive and negative yaw angles. The normal vector of the hyperplane was taken as the estimate of the yaw direction in the latent space. We collected 12,000 synthetic images for both positive and negative yaw angles except for angles in the interval $[-5°, 5°]$ to create some margin between the classes.

During training, the images were sampled as follows. For a particular identity, we independently randomly drew two yaw angles from the sequence, one for the hair image I_h and the other for the face image I_f. Thus, the input images have possibly nonmatching hair and face poses. The ground-truth image I has the yaw of the face image I_f. For clarity, the sampling is depicted in Figure 4.3.

Although the identity slightly changes in the sequence, the hairstyle stays the same. The identity loss is always computed between I_f and I, which have the same poses (I_f is extracted from I).

4.4 Data Augmentation

To achieve better results for input pairs with misaligned hair and face inputs, we used the following data augmentation. We first trained the network for 500,000 iterations with a random affine transformation of hair images I_h.

Figure 4.2: Examples of five-step sequence with changing yaw angle using learned direction in the latent space of the StyleGAN2 network.

Figure 4.3: An example of data sampling. The hair image I_h and I_f are independently randomly drawn from a 5-step yaw sequence, possibly having a mismatching pose. The ground-truth image I has consistent pose with the facial input I_f.

Namely, translation ±13px, scaling 0.85–1.15, and in-plane rotation ±20° was uniformly sampled. Face image I_f stayed the same.

After the training, we observed that the network was sensitive to a dislocation of the input face. Input face images having a slightly different position, scale, or in-plane rotation resulted in certain artifacts. Therefore, we fine-tuned the trained network with further 60,000 iterations with face images I_f transformed with the same affine transformation as the hair images, except for a smaller ±10° range of rotations. The same transformation was applied to the target images to preserve consistency. Note, that the affine parameters for I_h and I_f were sampled independently.

Interestingly, the network did not converge well when the inner face transformation was used from the beginning, and the resulting images suffered from artifacts. Typically, the contour of the hair leaked into the background. Two stage training solved the problem. The reason is probably that the network trained to capture a simpler transformation is able to guide the network to capture a more complex transformation.

4.5 Discussion

We tried multiple architectures to tackle the problem. One of the approaches was to encode I_f and I_h into a latent space of dimensions 1×512. However, the embeddings failed to capture the necessary detail to preserve the identity, and the resulting images were blurred. Another approach that we tried was to encode both the hair and face image jointly by concatenating them into a six-channel input. However, that approach did not allow independent control over hair and facial features. The best solution was to have two separate encoders for each of the inputs. The originally proposed architecture based on the feature pyramid in [22] was too large to fit twice into the GPU memory. Finally, we used a lighter encoder architecture with 18×512 dimensional output embeddings. However, employing two encoders still had an impact on the memory requirements. The GPU did not allow more than two samples per batch, mainly due to the architecture size itself and the additional networks used to compute the losses.

We discovered that the data augmentation in the form of affine transformation had a significant influence on the model performance. When we did not use the augmentation, the model produced specific artifacts when the input image was misaligned.

Enriching the dataset with nonfrontal poses and sampling of hair and face images with mismatching poses during training led to considerable improvement in hairstyle transfer for inputs with very different poses.

Chapter 5

Experiments

We present several experiments to evaluate the consistency and fidelity of the generated hairstyles. The qualitative analysis includes hairstyle transfer, hair synthesis for 3D morphable model [2], interpolation, latent space manipulation, and comparison with other methods for face swapping. The quantitative evaluation was performed by using an independently learned hair similarity metric. To quantitatively assess the visual quality of our results, we conducted a user study. All experiments were done on held-out test set which was not used in the training.

5.1 Qualitative Evaluation

5.1.1 Hairstyle Transfer

This experiment demonstrates the ability of the network to transfer hairstyles between different identities. The hairstyle transfer was executed on several image pairs by simply swapping the inner face inputs I_f.

In Fig.5.1 and 5.2, we present several examples of hairstyle transfer between a variety of identities and haircuts. The pairs are challenging, since the poses of the original images and of the hair inputs are different, as well as the illumination. The input identities differ in head size and shape, in gender and ethnicity. Some of the input haircuts are very complicated.

The hairstyles are faithfully transferred while still preserving the input identity and facial expression. The output images provide seamless hairstyle transfer despite the challenges. Note that, e.g., the skin color is well preserved and that even female hairstyles to male identities are transferred without apparent artifacts.

The reasons that the network can handle mismatching pairs of hair and inner face input images and produces appealing results are twofold. The robustness to geometrical misalignment is probably due to the complex data augmentation used in the training. The illumination consistency is probably due to the state-of-the-art StyleGAN2 generator embedded in our architecture. StyleGAN2 pretrained on a large amount of data produces photorealistic results with consistent illumination.

Figure 5.1: Hair transfer examples. The top box depicts the input images along with their auto-reconstructions. The bottom grid shows transfer of various hairstyles to different identities. The identities are preserved column-wise, the hairstyles row-wise.

5.1.2 Interpolation

To uncover the behavior of the latent spaces (of hair and faces), we experiment with interpolation.

A pair of images was taken, and the latent-space embeddings z_h, z_f of the respective hair and face encoders were found for both images. Then, we generated a sequence of several steps by linearly interpolating between either the latent codes of hair or face. We reconstructed the output images feeding the network with the interpolated codes of hair, while keeping the face code the same or vice versa. Note that the structure of our framework allows independent interpolation in both face and hair domains simultaneously.

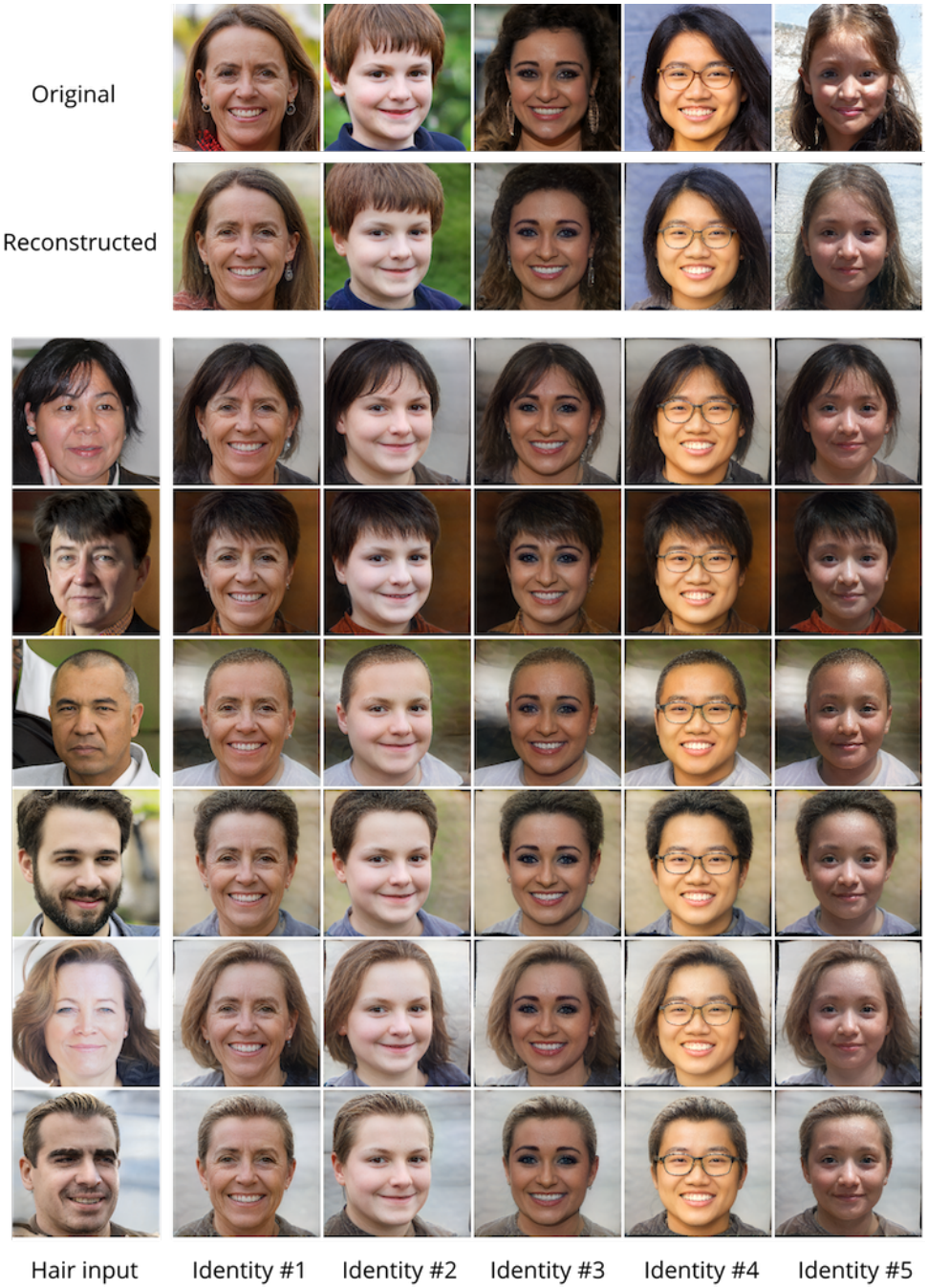


Figure 5.2: Hair transfer examples - additional results. The top row shows the original input, the row below depicts auto-reconstructed images. The grid displays hair transfer between various hairstyles and identities with diverse and challenging conditions.

Results for face interpolation are presented in Fig. 5.3. Face interpolation shows a gradual change in identity and lighting without modifying the hairstyle. The hair is naturally adjusted to fit a face size.

Hair interpolation is depicted in Fig. 5.4. The hairstyles are progressively changing length and color while preserving the identity. With closer observation, it is apparent that the color of the forehead is unified with that of the face, resulting in a realistic-looking portrait.

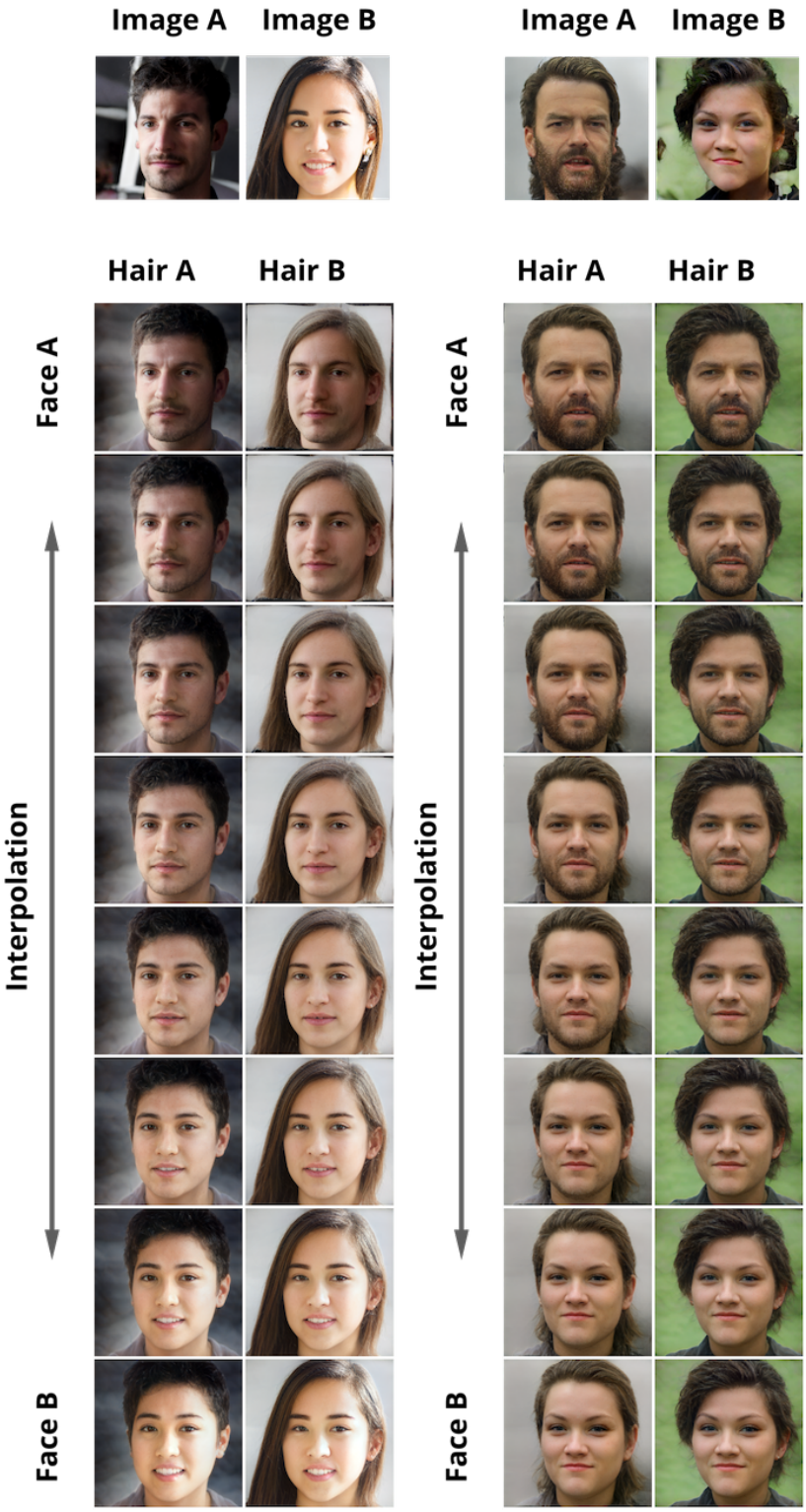


Figure 5.3: Interpolation in the face domain. The hairstyles are preserved column-wise. The top-most row depicts the input images A and B. The box below shows eight interpolation steps; from the top-row with identity from image A to the identity from image B in the bottom row.

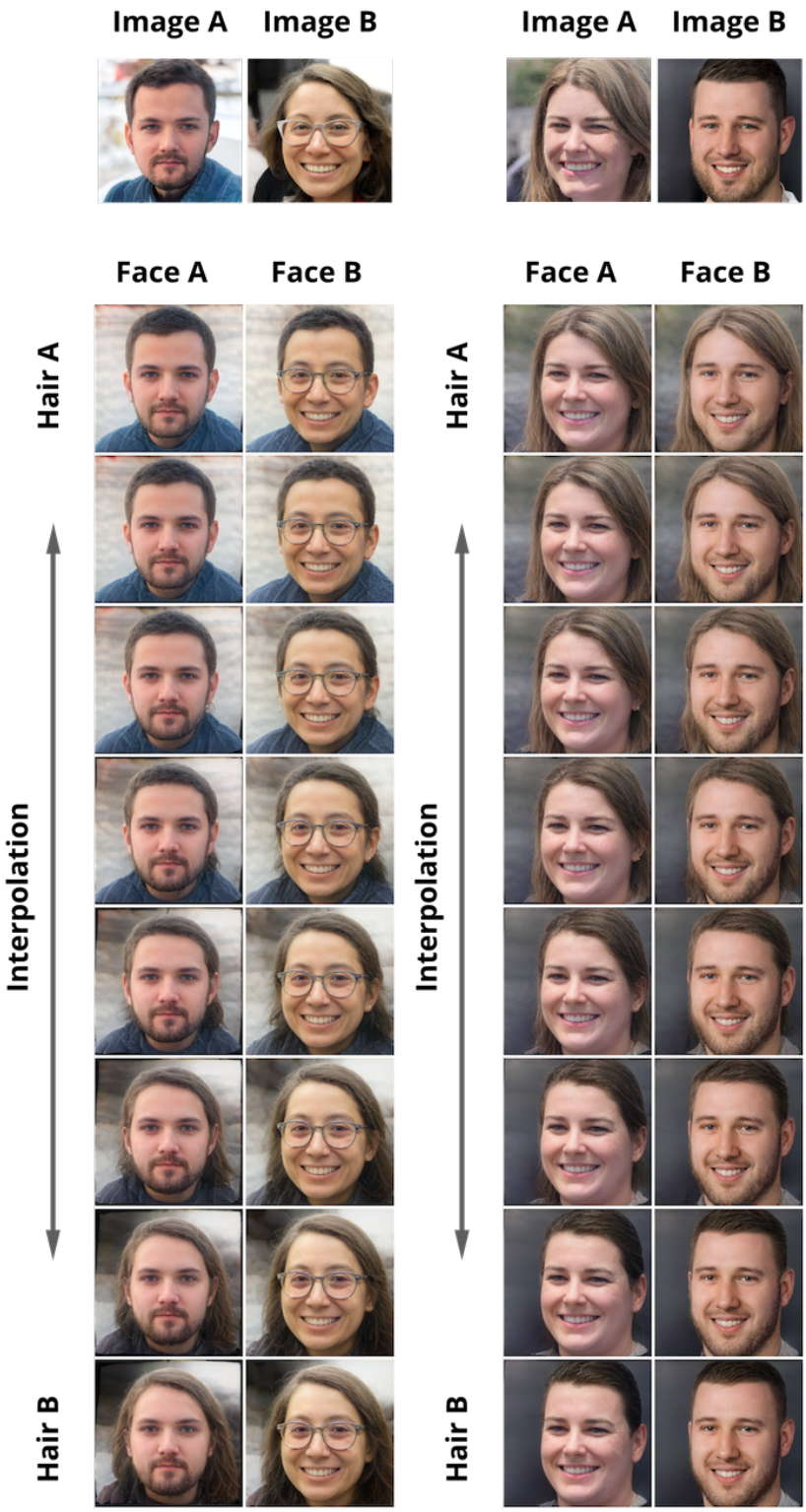


Figure 5.4: Interpolation in the hair domain. The identities are preserved column-wise. The top-most row depicts the input images A and B. The box below shows eight interpolation steps; from the top-row with hairstyle from image A to the hairstyle from image B in the bottom row.

5.1.3 3D Morphable Model

Since the introduction, the 3D morphable model (3DMM) has come a long way with increasingly detailed features. Nevertheless, the model is still missing hair. Our framework provides uncomplicated hair generation for faces rendered by 3DMM.

In Fig. 5.5a, we show two examples using the proposed hairstyle transfer method and the 3DMM. First, we fit a 3D morphable model to a test image and rendered a sequence of poses with increasing yaw angle. Each rendered image was masked to obtain the inner face image I_f. All images from the sequence were paired with the hair image I_h derived from the original test image. Those pairs were fed into the network to generate the output images. The Fig. 5.5b shows the same process with a randomly generated 3DMM. The rendered models and hair input image from the previous experiment were again fed into our network to provide a synthetic 3DMM model rendering with natural hair.

We can see that the hair rotates consistently with the input face. The identity is well preserved for all orientations. Only for extreme yaw angles, the output rotations seem slightly smaller than the inner inputs. The reason is probably a scarcity of extreme orientations in the training set. We only have images up to 25° in our datasets, since it was problematic to generate larger orientations without artifacts.

(a):

(b):

Figure 5.5: 3D morphable model with hair. The 3DMM was fitted to a real image (a), or generated randomly (b). A sequence of yaw angles in range $[-30°, 30°]$ was rendered. Hair was added from the real image.

5.1.4 Hair Manipulation

To explore the semantic properties of the latent space of the hair domain, we experimented with hair manipulation, namely, with hair color and hair structure.

For this task, we exploited an annotated dataset CelebAMask-HQ. We trained a linear SVM classifier to discriminate between latent codes for a given architecture.

For hair color manipulation, we collected images with positive annotation in the attributes *Blonde* and *Black*, creating two classes. The number of images in the black hair class was 7047 and in the blonde hair class 5622. The SVM penalty parameter C was set proportionally to adjust the disbalance between the classes. We extracted I_h from each image, as described in Section 4.1, and generated their latent codes z_h using the encoder E_h. The linear SVM gives the discriminating hyperplane between the two classes; its normal vector was used to manipulate the hairstyle in the latent space of E_h. The average accuracy in five-fold cross-validation was 99%.

Formally, the manipulation can be expressed as

$$z'_h = z_h + \mu n,$$

where z_h is the latent vector of dimension 18×512, $\mu \in \mathbb{R}$ is the magnitude of manipulation, and n is the semantic direction (i.e., the normal vector of the discriminating hyperplane of dimension 18×512). The vector z'_h is fed into the mapping network M along with the *fixed* facial embedding z_f. The results of the hair manipulation are depicted in Figure 5.6. It is visible that the color change does not affect other attributes like the background or the hairstyle itself.

The attributes regarding hair structure were *Wavy* and *Straight*. The *Wavy* class contained 7210 images, the *Straight* class 5459 images. The average accuracy in five-fold cross-validation was, in this case, only 83%. That is mainly because there is no fine line between straight and wavy hair; distinguishing between the two is hard even for a human annotator. The results of the hair manipulation are shown in Figure 5.7. The learned direction successfully changes the hair structure without changing the hairstyle much.

StyleGAN2 latent space manipulation is a popular technique that often provides appealing results for certain semantic directions, e.g., gender, age, pose [37]. Nevertheless, for larger magnitude of the linear manipulation, the results suffer from undesirable changes of identity, age and expression (e.g., Figure 4.2). We conducted the hair manipulation experiment described above with the StyleGAN2 latent space.

To obtain the latent vector for each annotated image, we employed an encoder trained for StyleGAN2 inversion from [22]. We used the same classes and a linear SVM classifier as described above. The estimated average accuracy in five-fold cross-validation was 99% for hair color and 80% for hair structure. The hair manipulation results are presented in Figures 5.8, 5.9. As

opposed to our method, hair manipulation in the latent space of StyleGAN2 changes shape of the face, identity, age, and expression. In some cases, the face does no longer look very realistic. That is probably caused by the fact that the latent space encodes all properties of the face and hair together.

Figure 5.6: Hair color manipulation in the latent space. The middle row shows the original image. The left column shows darkened hair, the right column shows lighter hair.

Figure 5.7: Hair structure manipulation in the latent space. The original image is depicted in the middle column. The left column shows straightened hair. Wavier hairstyles are depicted in the right column.

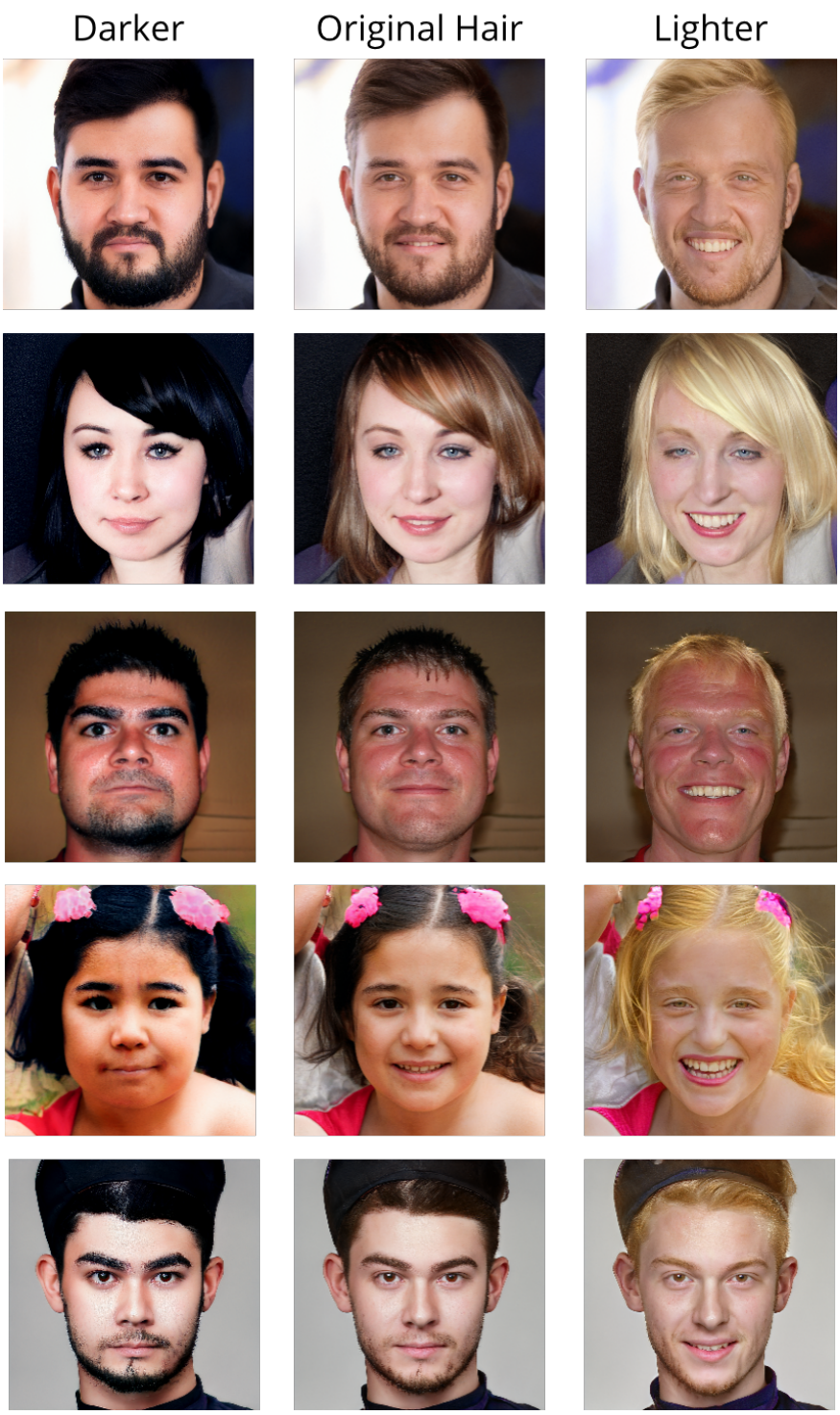


Figure 5.8: Hair color manipulation in the latent space of the StyleGAN2. The original image is displayed in the middle column. Each row shows the original image with darker hair color (left) and lighter hair color (right).

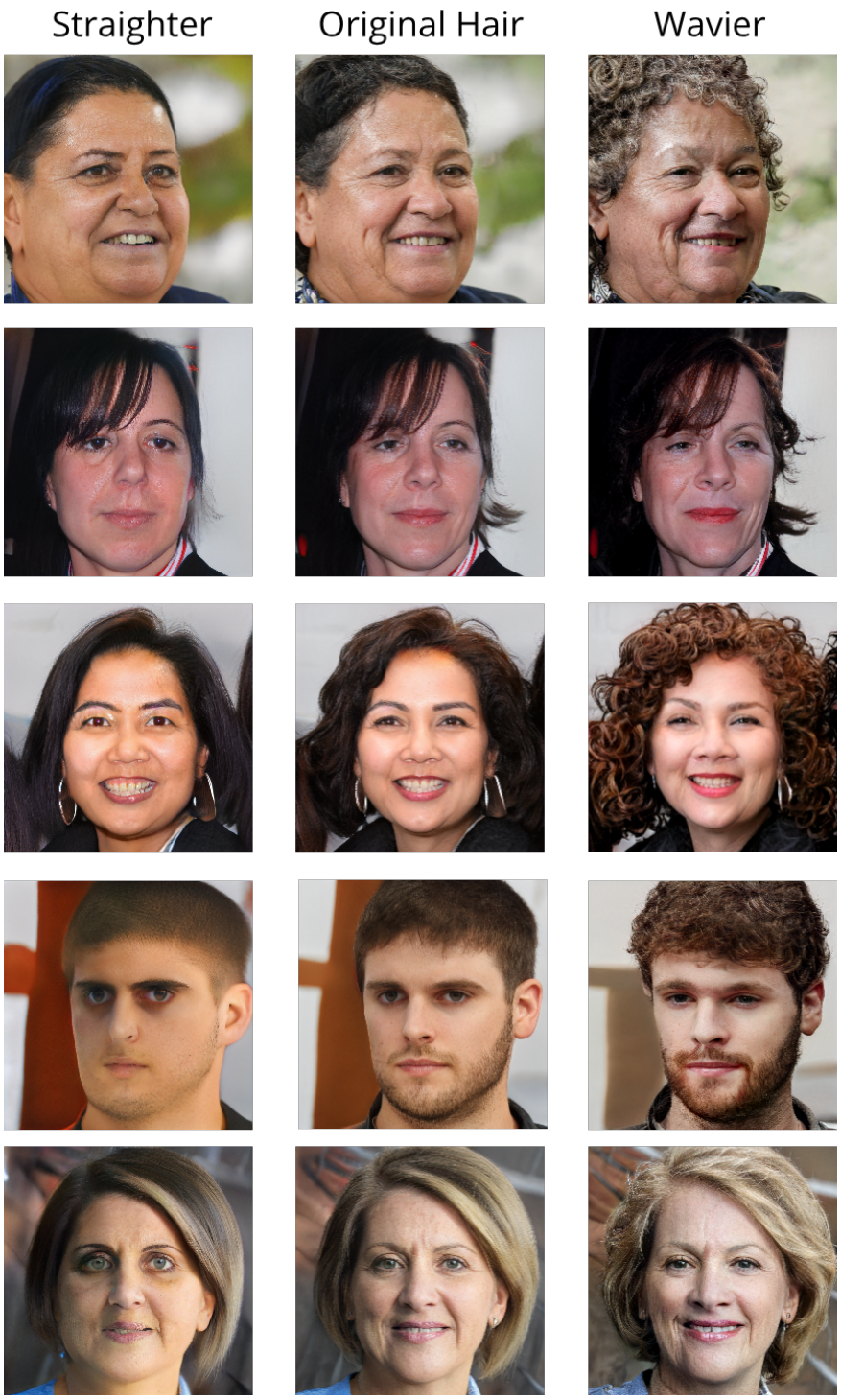


Figure 5.9: Hair strucutre manipulation in the latent space of the StyleGAN2. The original image is presented in the middle. Straighter hair is in the left column, wavier hair is in the right column.

5.1.5 Comparison with Other Methods

The closest works to ours are RSGAN [25], and MichiGAN [26]. The latter requires user interaction. Therefore, it cannot be easily compared—moreover, the background inpainter network is not available due to ownership rights. Comparison with RSGAN is shown in Fig. 5.10. The images are taken from the electronic version of the paper and then enriched by our results. This is a challenging scenario since we do not have the choice of the images in our hands. The RSGAN presents two methods for face swapping/hair transfer; RSGAN and RSGAN-GD. RSGAN is the output of their network, whereas RSGAN-GD reconstructs only the facial part and blends it with the input background and hair using gradient-domain image stitching [29]. We compare auto-reconstruction in Fig. 5.10 (a) and hairstyle transfer in (b) and (c). Note that our results in (b) have a different pose, since the pose is given by the source image and not by the target image. We simply fit the hairstyle to the input face and not the face to the hairstyle as other methods. This is given by our training process.

Our model achieves better results in reconstruction (a) and in preserving identity than RSGAN-GD, and Nirkin+ (b). Hairstyle fidelity in (b) is comparable among all images, nevertheless residuals of the original hair are visible in Shlizerman. In (c), we can see that our method outperforms other results, e.g. RSGAN fails to transfer long hair to male identity and vice versa.

5.2 Quantitative Evaluation

To evaluate the proposed method, we performed two additional experiments. The first experiment provides a quantitative evaluation of the hairstyle similarity between the input I and the generated image I_G. No hair masks were used to evaluate the measurements. The other experiment is a user study to assess the visual quality of the generated images of our method against random images from StyleGAN2 generator.

5.2.1 Hair Similarity Metric

To the best of our knowledge, there is no hairstyle similarity metric available. Therefore, we trained a hair similarity metric on an annotated dataset of hairstyles. Our method allows to train from databases with incompatible hairstyle annotations, which was exploited in our case.

Dataset

We combined two annotated datasets; **CelebAMask-HQ** [41] and a part of Hairstyle 30k [24] - "**Sixkind**". In summary, about 20k images.

CelebAMask-HQ. The dataset contains seven attributes regarding hair; *brown*, *black*, *blonde*, *gray*, *straight*, *wavy*, *bald*. We selected only images with

Figure 5.10: Comparison with other methods. Reconstruction (a), and hairstyle transfer/face swapping (b), (c). All images except for our results were adopted from [25]. Note that in (b) our method provides hair transfer results onto face image of the source image unlike the other methods. Compared methods are from Shlizerman [14], Nirkin [28], GD [29], VAE-GAN [38], ALI [39], α-GAN [40].

positive annotation in at least one of these attributes. The images with hats were discarded. Each image was assigned an attribute vector encoding the attributes found in the image. The images with the same attribute vector were considered as the same class. The total number of classes was 19.

Sixkind. Since some classes of the whole Hairstyle-30k dataset do not contain many examples, we chose only a subset of the dataset containing 6 classes. In each image, we detected the face with dlib detector [42]. We increased the size of each bounding box by a factor of 0.75 on all sides and shifted the larger box up by a factor of 0.5 (of the original bounding box size) to capture the whole haircut.

Training

We employed the architecture ResNet 50 without its last layer. The network was trained in Siamese setup (see Fig. 5.11) using triplet loss [43] with a

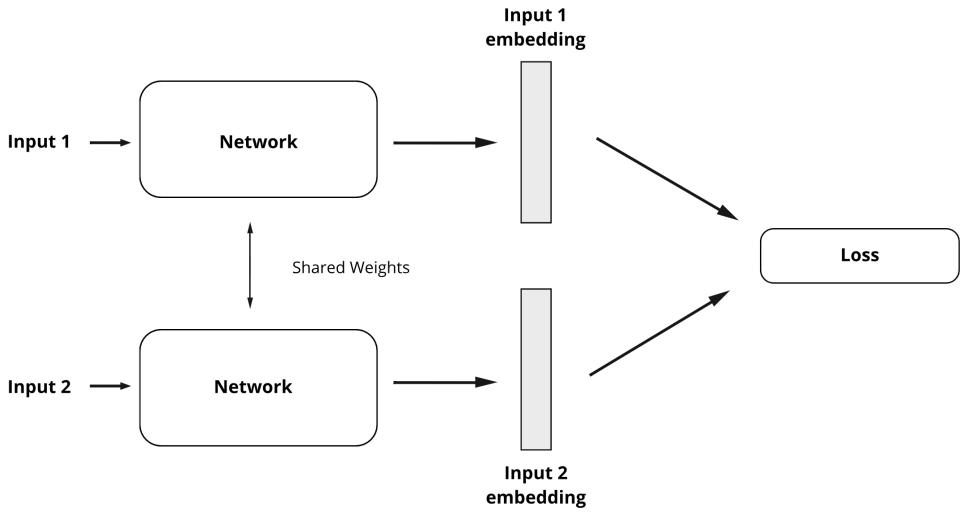


Figure 5.11: Siamese network setup. Two inputs are passed through the same network.

margin equal to 1 and L_2 distance. Formally,

$$\mathcal{L} = \max(||R(I_a) - R(I_p)||_2 - ||R(I_a) - R(I_n)||_2 + \alpha, 0),$$

where R denotes the network, I_a anchor image, I_p positive image, and I_n the negative image, α=1. Since the classes were not compatible across the datasets, each randomly drawn triplet contained only images from either dataset. We augmented each batch with "mirrored" triplets. These triplets were constructed from an anchor and a negative image; a positive example was created by horizontally flipping an anchor image. When feeding into the network, the images were resized to 224×224. We used the optimizer AdamW [44] with learning rate 0.0001 and weight decay 0.0001 for 300 epochs. Batch size was 32 with 8 more randomly chosen "mirror" triplets.

Retrieval Experiment

To visually verify the performance of the hair similarity metric, we carry out a retrieval experiment on a large database.

For all images I in the dataset, the embedding $R(I)$ is found by the trained hair similarity network. The L_2 distance is computed between the query embedding and all the embeddings of the dataset. Finally, the distances are ranked and the four nearest images are retrieved. The experiments were conducted on the held-out test set (with 5000 images) of the hairstyle dataset described above in Section 5.2.1 and on 100,000 random images generated by StyleGAN2.

The results are portrayed in Fig 5.12. The retrieved images indeed contain visually similar hairstyles and vary in pose and in identity.

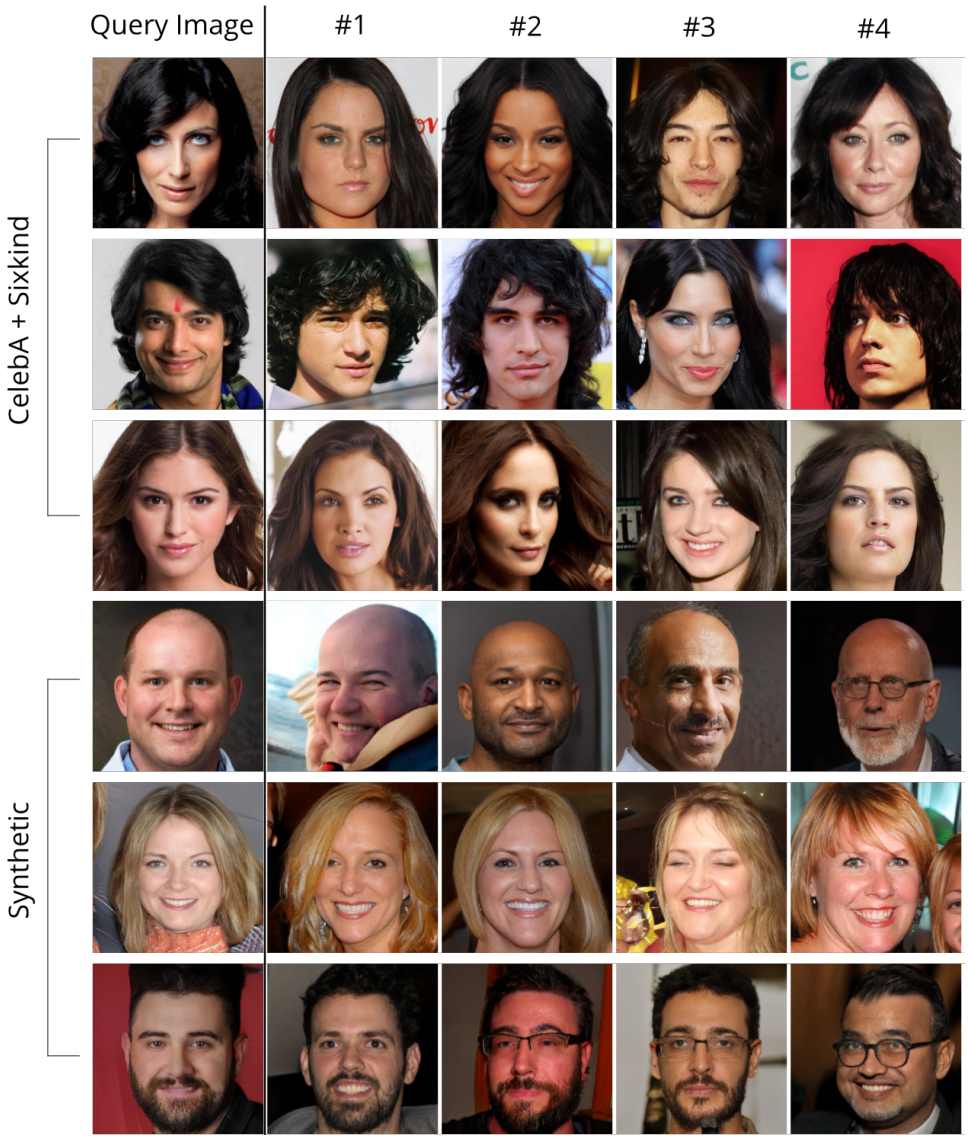


Figure 5.12: Retrieval experiment of the hair similarity metric. The left column presents the query image and the rest of the columns are the closest images in a sense of hair similarity. Three upper rows are from the test set of the Hairstyle dataset. The lower rows are from the dataset generated by StyleGAN2.

ROC curves

The learned hair similarity metric was used to evaluate the hair transfer fidelity. In Fig. 5.13, we show ROC curves for a binary classifier which uses the similarity metric to distinguish pairs of images with the same or different hairstyle. We compare the results for image pairs generated from the real dataset with manually labeled hairstyles and synthetic pairs generated by our hairstyle transfer method.

It is seen that the ROC curves are indistinguishable. The metric learned from manually created labels simulates the answer of a human to the question: "Are the two hairstyles similar?". Hence, we see that a human would agree with our method as frequently as he/she agrees with another human on what pair of images shows the same hairstyle. This evaluation technique replaces an expensive user study which would require to manually judge all image pairs generated by our method.

The positive and negative image pairs were generated as follows. Pairs for the manually labeled dataset were constructed using the class labels. ROC curve is computed from 2500 random pairs from held-out test sets.

A synthetic set of images generated by our method was prepared as follows. Let us have a pair of images A and B, randomly generated from StyleGAN2. Then we denote $\mathcal{G}(A_h, B_f)$, the output of the proposed hairstyle transfer network taking the hair input from A and the face input from B. For every pair (A, B) we create two positive $\{\mathcal{G}(A_h, A_f),\ \mathcal{G}(A_h, B_f)\}$, $\{\mathcal{G}(B_h, B_f),\ \mathcal{G}(B_h, A_f)\}$ and two negative pairs $\{\mathcal{G}(A_h, A_f),\ \mathcal{G}(B_h, A_f)\}$, $\{\mathcal{G}(B_h, B_f),\ \mathcal{G}(A_h, B_f)\}$. We generated 2500 test pairs as in case of the real dataset.

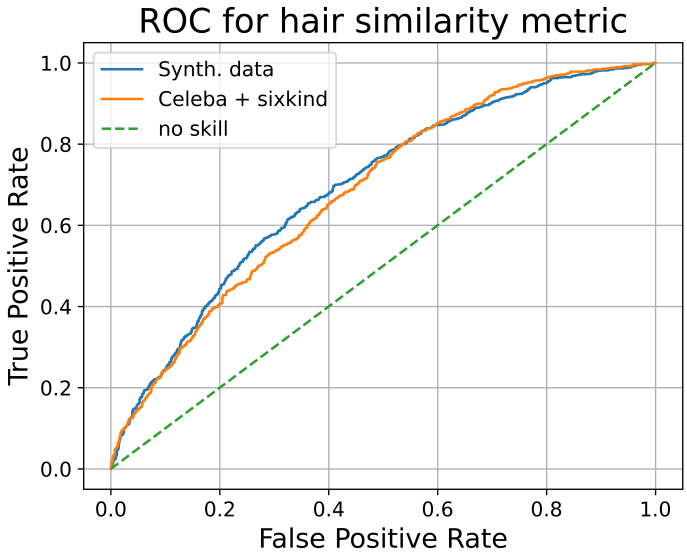


Figure 5.13: ROC curves for hair similarity metric computed on a manually labeled dataset (orange) and an unannotated synthetic dataset generated by our hairstyle transfer method (blue).

5.2.2 User Study

To quantitatively analyze the visual quality of the hair-transferred results, we conducted a human-evaluated survey. The goal of the study is to compare our results with raw images generated by StyleGAN2.

We set up four questionnaires, all containing 25 pairs. Each pair comprised a randomly generated image by StyleGAN2 and an image generated by our method with the same identity but a different hairstyle, randomly chosen from the image set. The question was: "Which of the two images is more realistic?". All images captured only women to avoid unrealistic results due to unusual hairstyles from male to female hair transfers.

Each respondent was assigned a score, representing the number of votes for the image generated by our method. The histogram of the scores across all surveys is depicted in Figure 5.14. A total of 119 respondents participated in the study. The mean score is 46%, standard deviation 15.3%. The mean score suggests that 46% of the hair-transferred images were more realistic than the images generated by the StyleGAN2. The best theoretical score is 50%, since we use the StyleGAN2 as an output generator. The result suggests that our method is not easily distinguishable from the StyleGAN2. It means that the image quality perceived by study participants is comparable as that of the state-of-the-art StyleGAN2 and the hair transfer does not introduce additional artifacts. In the Figure 5.15a, we present several pairs that got consistently voted for StyleGAN2. The generated images look unnatural due to anomalies in the source images, e.g., a hat or unnatural hair color. In contrast, the Figure 5.15b portrays some pairs that were frequently voted for our hair transfer results.

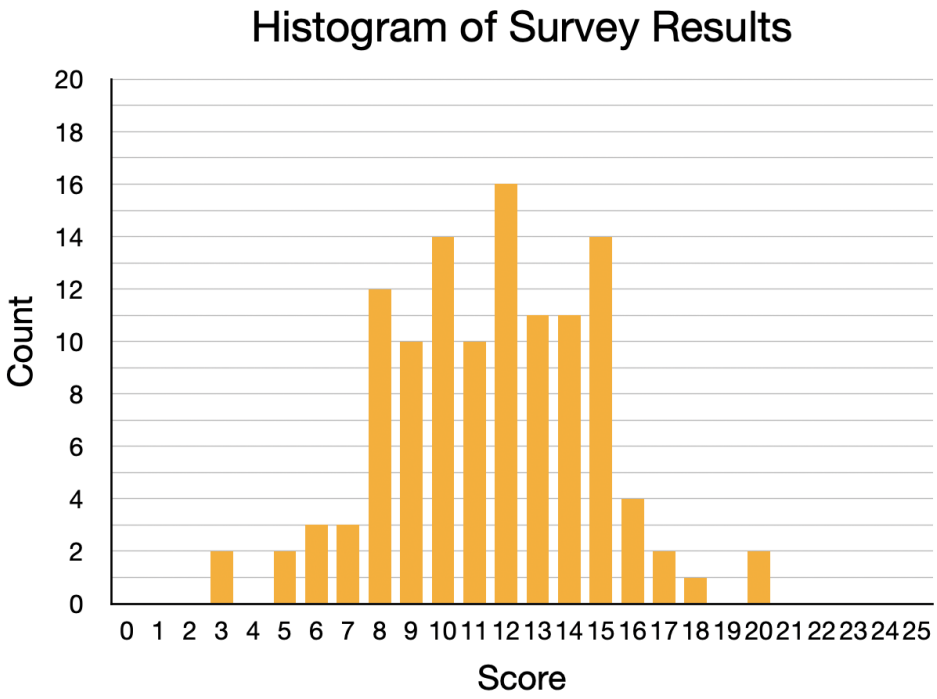
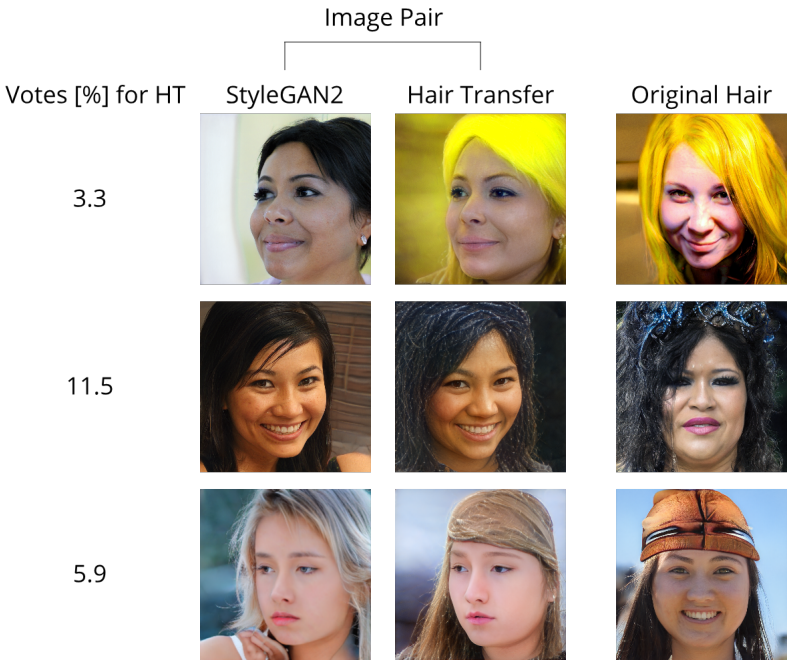
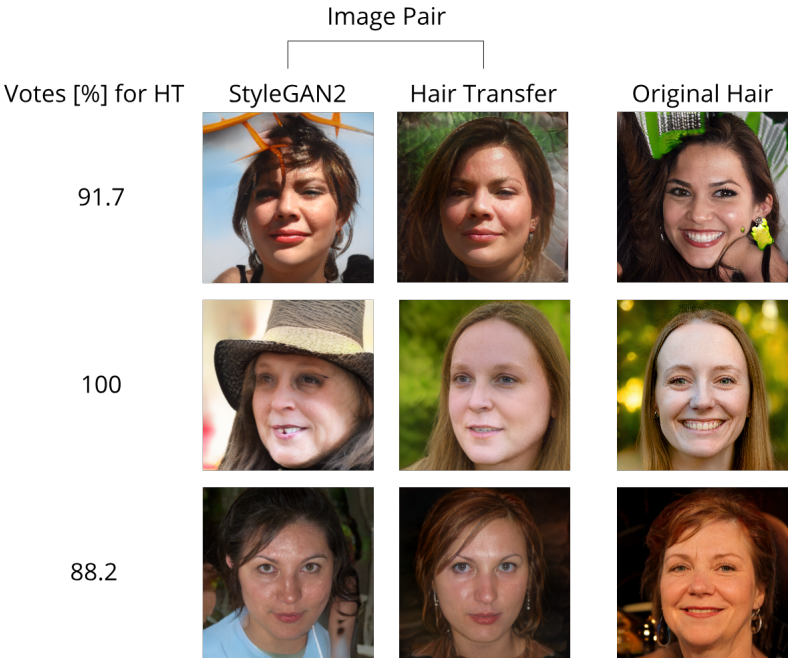


Figure 5.14: The histogram of scores in conducted user study. The horizontal axis represents acquired score; number of selected images generated by our method. The vertical axis describes number of respondents with a given score.



(a) : Failure cases in hair transfer.



(b) : Success cases in hair transfer.

Figure 5.15: Example pairs from the user study. Failure cases (a) were frequently voted for StyleGAN2, while the success cases (b) were consistently voted for the hair transferred images (HT). The left-most column displays percentage of votes for the hair-transferred image. The two columns were presented in the questionnaires as an image pair. The right-most column shows the hair source for the hair transfer.

Chapter 6

Conclusion

This thesis presented a method that provides a high-definition hairstyle transfer between face images with additional applications, e.g. adding hair to an image rendered by 3D morphable model. The method evades the intricate GAN training and does not require any annotations or even an external dataset. The training is accomplished on a fully synthetic dataset generated by the StyleGAN2.

The framework employs two encoders, one for hair and one for face, a mapping network, and *fixed* StyleGAN2 generator. The method was thoroughly evaluated using both qualitative and quantitative analysis. Specifically, we presented experiments on hairstyle transfer, hair generation for a rendered 3D morphable model, and semantic hairstyle manipulation. We further assessed the hairstyle transfer performance using the learned hair similarity metric and the user study. Furthermore, we compared our method to other state-of-the-art methods. The results of our method show high fidelity of hair transfer, preserving the identity and facial expressions of the subjects. Good results are achieved even in challenging conditions such as hairstyle transfer for pairs with different poses, ages, ethnicities, and lighting conditions.

Unlike other methods, we do not use any post-processing of the results, e.g., blending the original background with the output [25]. Our pipeline is simple and fully automatic, unlike [26], which requires user interaction. The results are provided by a single pass of the proposed neural network without iterative optimization. Furthermore, the output resolution attains 1024×1024 pixels, which exceeds other hairstyle manipulation methods.

The performance could probably be further improved by segmentation of the background. The background is not relevant in hair transfer application, but the hair encoder is forced to learn it, which can considerably limit the effective capacity.

Appendix A

CD Contents

- HairstyleTransferbetweenPortraitImages_subrtade.pdf – the thesis itself
- thesis.zip – LaTex project of this thesis
- figures/ – folder containing some of the images in the original resolution
- HairTransferVideo.mp4 – a video demonstrating applications of the proposed framework

Appendix B

Bibliography

[1] Antl-Weiser Walpurga. The time of the willendorf figurines and new results of palaeolithic research in lower austria. *Anthropologie (Brno)*, 47(1-2):131–141, 2009.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, 1999.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proc. CVPR*, 2020.

[5] Adéla Šubrtová, Jan Čech, and Vojtěch Franc. Hairstyle transfer between face images. In *2021 IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, 2021. In Review.

[6] Sylvain Paris, Will Chang, Oleg I. Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair photobooth: Geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.*, 27(3), 2008.

[7] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *Proc. of CVPR*, 2012.

[8] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.*, 31(4), 2012.

[9] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. Dynamic hair manipulation in images and videos. *ACM Trans. Graph.*, 32(4), 2013.

[10] Yanlin Weng, Lvdi Wang, Xiao Li, Menglei Chai, and Kun Zhou. Hair Interpolation for Portrait Morphing. *Comput. Graph. Forum*, 2013.

[11] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Trans. Graph.*, 33(4), July 2014.

[12] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.*, 34(4), 2015.

[13] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Trans. Graph.*, 35(4), July 2016.

[14] Ira Kemelmacher-Shlizerman. Transfiguring portraits. *ACM Trans. Graph.*, 35(4), July 2016.

[15] K. Ward, F. Bertails, T. Kim, S. R. Marschner, M. Cani, and M. C. Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Trans. Vis. Comput. Graphics*, 13(2), 2007.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. of ICLR*, 2016.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. of ICLR*, 2018.

[18] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proc. of ICLR*, 2019.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019.

[20] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proc of ICCV*, 2019.

[21] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 38(4), 2019.

[22] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and D. Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *ArXiv*, abs/2008.00951, 2020.

[23] Richard Zhang, Phillip Isola, Alexei A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF CVPR*, 2018.

[24] Weidong Yin, Yanwei Fu, Yiqing Ma, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. Learning to generate and edit hairstyles. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.

[25] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Rsgan: Face swapping and editing using face and hair representation in latent spaces. In *ACM SIGGRAPH 2018 Posters*, SIGGRAPH '18, 2018.

[26] Zhentao Tan, M. Chai, Dongdong Chen, Jing Liao, Q. Chu, Lu Yuan, S. Tulyakov, and Nenghai Yu. Michigan: multi-input-conditioned hair image generation for portrait editing. *ACM Trans. Graph.*, 39, 2020.

[27] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumer, and S.K. Nayar. Face swapping: automatically replacing faces in photographs. *ACM Trans. Graph.*, 2008.

[28] Yuval Nirkin, Iacopo Masi, Anh Tuan Tran, Tal Hassner, and Gérard G. Medioni. On face segmentation, face swapping, and face perception. *CoRR*, abs/1704.06729, 2017.

[29] Anat Levin, A. Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *ECCV*, 2004.

[30] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3), 2003.

[31] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

[32] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *ArXiv*, abs/1703.10717, 2017.

[33] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. of ICCV*, pages 1501–1510, 2017.

[34] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017.

[35] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017.

[36] Jiankang Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *Proc. CVPR*, 2019.

[37] Yujun Shen, Jinjin Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. *2020 IEEE/CVF CVPR*, 2020.

[38] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

[39] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. *ArXiv*, abs/1606.00704, 2017.

[40] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks. *ArXiv*, abs/1706.04987, 2017.

[41] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF CVPR*, pages 5549–5558, 2020.

[42] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(60), 2009.

[43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE CVPR*, pages 815–823, 2015.

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.