AGE AND GENDER NORMALIZATION IN KINSHIP VERIFICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Oğuzhan Çalıkkasap
September 2021

Age and Gender Normalization in Kinship Verification
By Oğuzhan Çalıkkasap
September 2021

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Hamdi Dibeklioğlu(Advisor)

Selim Aksoy

Pınar Duygulu Şahin

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

AGE AND GENDER NORMALIZATION IN KINSHIP VERIFICATION

Oğuzhan Çalıkkasap
M.S. in Computer Engineering
Advisor: Hamdi Dibeklioğlu
September 2021

Kinship verification from facial images using deep learning is an interesting problem that is unsolved and gains growing attention of the research community. However, the most recent kinship verification systems suffer from age- and gender-related facial attributes that cause problems in kinship verification between subjects of different age and gender.

In this study, we propose various methods to reduce the negative effect of the age- and gender-related facial attributes in kinship verification to achieve a more robust verification model. The proposed approach utilizes the comprehensive modeling capabilities of the recent generative adversarial network architectures to model the age and gender of subjects and reduce their effect in kinship verification, if not remove entirely. Furthermore, we conduct a thorough analysis over individual and combined effects of age and gender normalization, performed in both image and latent space of the generative models. Lastly, we investigate the impact of additional emphasis on the facial identity information during the normalization process.

Taking one of the most recent kinship verification models as our baseline, we show that gender normalization has reduced the verification performance gap between subject pairs with the same and different gender, up to 6%. Furthermore, joint normalization of age and gender improves the kinship verification accuracy up to 5% and 10% on two different in-the-wild kinship datasets. Therefore, this thesis proposes generic approaches to improve the reliability and robustness of kinship verification by normalizing the age and gender attributes without making changes in the core architecture of the employed kinship verification system.

Keywords: kinship verification, generative modeling, age and gender normalization.

ÖZET

AKRABALIK DOĞRULAMASINDA YAŞ VE CİNSİYET NORMALİZASYONU

Oğuzhan Çalıkkasap
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Hamdi Dibeklioğlu
Eylül 2021

Derin öğrenme kullanarak yüz görüntülerinden akrabalık doğrulaması, çözülmemiş ve araştırma topluluğunun artan dikkatini çeken ilginç bir problemdir. Bununla birlikte, en yeni akrabalık doğrulama sistemleri, farklı yaş ve cinsiyete sahip denekler arasında akrabalık doğrulamasında sorunlara neden olan yaş ve cinsiyete bağlı doğuştan gelen yüz özelliklerinden muzdariptir.

Bu çalışmada, daha sağlam bir doğrulama modeli elde etmek hedefiyle akrabalık doğrulamasında yaş ve cinsiyete bağlı yüz özelliklerinin olumsuz etkisini azaltmak için çeşitli yöntemler öneriyoruz. Önerilen yaklaşım, deneklerin yaşını ve cinsiyetini modellemek ve tamamen ortadan kaldırmasa da akrabalık doğrulamasındaki etkilerini azaltmak için güncel üretken çekişmeli ağ mimarilerinin kapsamlı modelleme yeteneklerini kullanır. Ayrıca, üretici modellerin hem imge uzayı hem de öğrenilmiş uzayında gerçekleştirilen yaş ve cinsiyet normalizasyonunun bireysel ve birleşik etkileri üzerinde kapsamlı bir analiz yapıyoruz. Son olarak, normalizasyon sürecinde yüz kimliği bilgilerini daha fazla vurgulamanın etkisini araştırıyoruz.

En yeni akrabalık doğrulama modellerinden birini temel alarak, cinsiyet normalizasyonunun, benzer ve farklı cinsiyetteki akrabalar arasındaki doğrulama performansı farkını %6' ya kadar azalttığını gösteriyoruz. Ayrıca, yaş ve cinsiyetin ortak normalizasyonunun, iki farklı akrabalık veri setinde akrabalık doğrulamasını %5 ve %10'a kadar arttırdığını gösteriyoruz. Bu nedenle bu tez, kullanılan akrabalık doğrulama sisteminin çekirdek mimarisinde değişiklik yapmadan yaş ve cinsiyet özelliklerini normalize ederek akrabalık doğrulamasının güvenilirliğini ve sağlamlığını geliştirmek için genel yaklaşımlar önermektedir.

Anahtar sözcükler: akrabalık doğrulaması, üretken modelleme, yaş ve cinsiyet normalizasyonu.

Acknowledgement

I sincerely thank my supervisor Hamdi Dibeklioglu for his guidance throughout my master's studies with his friendly approach in exposing various aspects of conducting a proper research. I have always felt like a team and learned a lot in each of our weekly meetings.

I would like to mention that I was so lucky to get to know amazing people during my first year in Ankara, such as Burak Mandira, Dersu Giritlioglu, and Diala Erekat, who are my fellow researchers. I believe this period of our academic lives will be remembered with common feelings like pain, sweat, and you-know-why not-so-frequent nightly hangouts.

Thanks to the best neuroscientist I have ever known, Kerem Kurban, for his endless company during our up-all-night studies, next-door coffee breaks, random discussions on arbitrary topics during my stay in 79, and even my second year in Switzerland. Also, he was the reason that I made another amazing friend who is Yunus Emre Koc.

I also appreciate getting to know such great friends like Luís Espírito Santo and André Carreira. Our backyard parties, Crossfit competitions, and D&D sessions made my time at CERN even more fun.

I am so grateful to my parents Gulay and Vedat, and my brother Burak for always being there when I needed them and their endless support on anything I strive for. Without them, there would not be such an achievement as today.

Last but not least, I am so filled with gratitude to have a wonderful spouse like Aysegul, who helped to tackle any difficulties that I faced during my studies. She showed an extraordinary understanding, consistently motivated me from the beginning to the end, and helped me in any possible way that she could. I honestly believe that half of this master's degree belongs to her.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Kinship means that having a blood relationship between humans, which indicates that a variety of genetic similarities can be observed. Considering the facial attributes, these inheritances might be observed as the similarity between a number of facial parts such as nose, mouth, etc. These similarities in the appearance of two different faces constitute clues for kinship analysis that humans perform visually during their daily lives without even noticing. For instance, when we see an adult man and a boy sitting next to each other, we can decide whether the boy is the son of that man based on their physical appearance. Various studies have been conducted under different contexts to understand the human way of kinship verification from facial images [1, 2, 3], where subjects are asked to determine whether the people in a given pair of facial portrait images are kin or not. In order to perform such an analysis, humans utilize different features, including the color of eyes, hair, and facial appearance in general [4].

There are different potential application areas for kinship verification including but not restricted to organization of large collections of family pictures, resemblance recognition of humans, surveillance, social media analysis, or finding missing family members on the internet. Subsequently, kinship information is quite valuable from diverse aspects. Although biometric verification methods like DNA tests provide almost an accuracy of 100 percent, they are more costly,

time-consuming, and sometimes not applicable. Therefore, automatic kinship verification from facial images is an interesting area of research which gains increasing popularity.

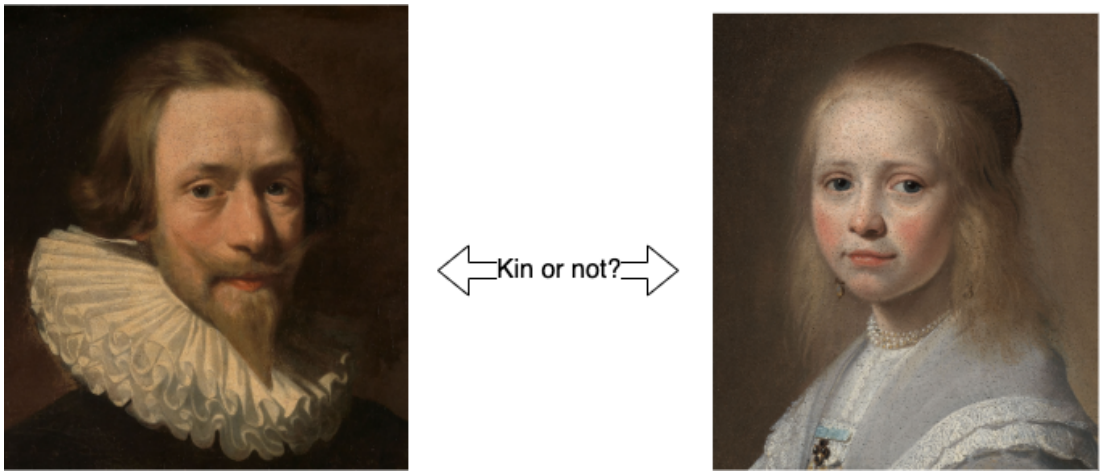


Figure 1.1: Kinship verification from facial images

The substantial variability across different kin pairs and kinship relations makes this research topic particularly difficult. As the primary goal of this field's study has been to recognize kinship traits in images automatically, deep learning models showed significant progress in recent years by reaching up to 88 percent of verification accuracy in the commonly used kinship verification datasets [5]. However, most kinship verification studies point out the imbalanced verification performance between the same and opposite gender pairs [6, 7, 8, 9], where the verification of the pairs with the same gender outperforms the verification of pairs with the distinct gender. Although there is a pretty limited number of complementary studies on the effect of age in kinship verification [10] due to the lack of age labels of kinship verification datasets, we hypothesize that the age gap of the pairs causes a similar artifact in kinship verification. We investigate the undesirable impact of the age and gender attributes and propose several methods to eliminate this drawback in kinship verification in a model-agnostic manner.

1.1 Motivation

The latest research on kinship verification focuses on developing more sophisticated and specialized deep learning model architectures and similarity metrics in terms of enhancing the kin-related feature extraction quality for a better verification performance. However, there is a very limited number of work focused on analyzing the inherent effects of age and gender in kinship verification, which introduces a number of side-effects especially considering the age gap or gender dissimilarities between the pairs that are to be verified.

In this study, we focus on analyzing and removing the effect of age- and gender-related facial attributes in kinship verification. Exploiting the comprehensive modeling capabilities of the recent generative adversarial networks, we model the age and gender-specific facial attributes and minimize their effect in kinship verification in order to obtain a more robust system against these specific attributes.

1.2 Kinship Verification

Studies in kinship analysis can be divided into two main branches that are kinship verification and kinship identification. Kinship verification aims at determining whether the people in a given pair of facial portrait images are kin or not. On the other hand, Kinship identification tries to figure out the exact kinship relation between two individuals. Since this study is solely focused on kinship verification, the following chapters will mainly cover the verification-oriented topics and provide complementary information from the identification domain when necessary.

The pace of study in utilizing deep learning methods in kinship verification has accelerated since the release of several kinship verification datasets, such as KinFaceW-I [11], KinFaceW-II [12], TSKinFace [13], Family-101 [14], CornellKin [15], UB-KinFace [16], and UvA-NEMO [17]. KinFaceW-I and KinFaceW-II features facial portraits of four kin relationships that are Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D), which

are collected from unconstrained images. Facial images in KinFaceW-I differ from KinFaceW-II as they are collected mostly from separate images. In tri-subject kinship face database TSKinFace, kin pairs are grouped in the child-parents format, including the both parents and their child as a kin group. Family-101 has a family structure consisting of 101 different family trees and 607 individuals. CornellKin consists of 143 pairs of parent-child kin pairs who are mostly the public figures or celebrities collected from the internet. UB-KinFace contains children and their parents at various ages, comprised of 600 images of 400 people having the same kin pairs as in KinFaceW datasets. Lastly, UvA-NEMO is distinguished from the prior image datasets, involving spontaneous and posed smile video footage of 400 subjects. Therefore, UvA-NEMO is rather used for kinship verification as taking the temporal cues into account as well. From among the datasets mentioned above, KinFaceW-I and KinFaceW-II are more widely used in image-based kinship analysis, thus we choose to use these datasets in our study.

The path of automatic kinship recognition starts with the work of Fang et al. [15], in which facial characteristics such as skin color, location and form of face components, and gradient histograms are used to verify kinship. They report a kinship verification accuracy of 70.67% on the dataset they introduce that is CornellKin.

A transfer subspace method for kinship verification is proposed by Xia, Shao, and Fu [10], aimed at reducing the age gap between the children and parents by simply using the younger image of the parents as an intermediate distribution between the young children and old parent pairs. Their method achieves an accuracy of 60% in kinship verification on the UB-KinFace dataset, which is released along with their work.

Building upon [10], Shao, Xia, and Fu [16] utilize Gabor filters alongside metric learning and transfer subspace learning. They enforce the gap between real kin pairings to a minimum while the distance between non-kin pairs is kept at a maximum. Reporting on their newly released dataset UB KinFace v2, they show a verification accuracy of around 69%.

Yan et al. [18] propose multiple metric learning methodologies such as multiple distance metrics are learned using different facial descriptors that encode various aspects of the face characteristics. Along with the likelihood of kin images belonging together, the correlation of different characteristics of the same sample is augmented. Authors validate their method on other verification datasets, including KinFaceW-I, KinFaceW-II, CornellKin, and UB-KinFace.

Xu and Shang [19] introduce another metric-based verification method for kinship analysis. They concurrently learn multiple sparse bilinear similarity models, utilizing sparsity-inducing norms that are formed as a joint structure. They enhance the gap in terms of the similarity metric between the non-kin pairs while minimizing it for the kin pairs as they use the interactions and correlations between the multiview data to generate fused and higher-level information. Authors report an improved verification performance compared to the prior multimetric-learning-based methods.

A distance-based hybrid approach is proposed by Mahpod and Keller [20], in which a multiview combined symmetric and asymmetric distance learning network is trained for kinship verification. In this method, authors formulate kinship verification as a classification problem where they employ the support vector machines to solve this classification task. A margin maximization learning technique is used to train dual discriminative representations for parents and children. The method is tested on KinFaceW and CornellKinFace databases, showing comparable performance to the earlier state-of-the-art methods.

Several studies have attempted to create strong face representations to obtain a better verification. Zhou et al. [21] exploits a spatial pyramid learning-based (SPLE) feature descriptor that combines the local appearance and global spatial information for a comprehensive representation of the facial attributes and applies support vector machines on top of that representation for kinship verification. The results show comparable performance to the human observers on the dataset collected for this study by the authors.

In their work Guo and Wang [22] suggest adapting the DAISY descriptors

in kinship verification to better express the essential features while a dynamic method for stochastically combining family attributes is being developed. They compare their verification results against [15] and report an improvement using this baseline.

Zhou et al. [23] propose another feature descriptor for facial representation that is so-called the Gabor-based Gradient Orientation Pyramid (GGOP). Authors apply a Gabor wavelet to each face picture to generate a series of Gabor magnitude (GM) feature images at various scales and orientations. Then for kinship verification, SVM is used on top of the extracted Gradient Orientation Pyramid (GOP) features from each GM feature picture. A performance in kinship verification comparable to the human observers is noted in their experimental results.

Kohli, Singh, and Vatsa [6] uses the local description, or the self-similarity representation, of the pre-processed Weber face image to perform kinship classification. After extracting the key points from the normalized facial images, self-similarity descriptors are obtained. Like in the previous works, an SVM classification head is employed on top of the descriptors to detect kinship's presence.

By utilizing the hierarchical local regions, Xia, Shao, and Fu [24] derive binary characteristics such as being male or female, along with the technique based on attributes is employed for creating middle-level representations such as having bigger or smaller eyes. After combining these attributes, kinship verification is performed by an SVM classifier.

Puthenputhussery et al. [25] propose a SIFT flow-based genetic vector feature extraction for encoding the kinship-oriented facial features. The authors point out the intuitive resemblance between the extracted genetic markers and the anthropological results in the literature. They intend to increase the similarity of parent and child features based on SIFT flow and learn an inheritable Fisher vector feature. These features are then evaluated by using a similarity metric that is the fractional power cosine similarity. The performance of the approach is validated on KinFaceW datasets.

Method	Algorithm
Fang et al. [15]	Gradient orientation pyramid on Gabor features
Xia, Shao, and Fu [10]	Transfer subspace learning
Shao, Xia, and Fu [16]	Metric learning using Gabor filters
Yan et al. [18]	Multiple distance metric learning
Xu and Shang [19]	Multiple sparse bilinear similarity modeling
Mahpod and Keller [20]	Multiview distance learning
Zhou et al. [21]	Spatial pyramid learning-based feature extraction
Guo and Wang [22]	DAISY descriptor extraction and SVM classifier
Zhou et al. [23]	Gabor-based Gradient Orientation Pyramid
Kohli, Singh, and Vatsa [6]	Weber face self-similarity representation
Xia, Shao, and Fu [24]	Hierarchical local regional features
Puthenputhussery et al. [25]	SIFT flow based genetic vector feature extraction
Zhang, Song, and Qi [26]	Deep convolutional neural network
Dehghan et al. [27]	Gated autoencoders
Wang et al. [28]	Stacked autoencoders for project space learning
Li et al. [29]	Convolutional Siamese networks

Table 1.1: Overview of kinship verification methods

Recent research has begun to use deep architectures as a result of substantial advancements in deep learning. Unlike the traditional methods, which extract facial features using manually designed descriptors, Zhang, Song, and Qi [26] propose an end-to-end deep convolutional neural network model to extract high-level facial features for kinship verification. These features are fed into the final layer, where a softmax classifier determines the kinship score.

To distinguish parent-offspring relationships, Dehghan et al. [27] use gated autoencoders to merge produced characteristics with a discriminative neural layer at the end. In essence, the relationship between the input pair of pictures is

learned using a gated autoencoder-based generative model. Following the generative modeling, discriminative training is performed in order to determine if the input pair pictures are kin.

Wang et al. [28] utilize stacked autoencoders to learn non-linear features followed by a metric learning approach. They first extract the facial features from the image pairs and feed them into a cascaded architecture of autoencoders, where the latent space of an autoencoder is the input of the next one. Then these representations are stacked, and metric learning is applied to find an appropriate project space such that the distance is smaller when the input pair is kin, while vice-versa is valid for the non-kin input. Note that the decoder part of the autoencoders is removed after the training phase. Authors validate their approach on KinFaceW-I and KinFaceW-II datasets and report an overall verification accuracy of 66.9% and 71.3%, respectively.

Li et al. [29] propose an approach where they train a convolutional Siamese network with architecture-specific constraints employing a similarity metric. An input pair of images are first fed into two convolutional neural networks that share weights, and the L-1 distance of both network outputs is computed. This distance is then used to compare against a learned threshold to obtain the final kinship score. The method is validated on the KinFaceW datasets and showed certain improvements in the verification of different kin pairs.

An overview of all the mentioned kinship verification algorithms are summarized in Table 1.1.

1.3 Effect of Age and Gender in Kinship Verification

There is a minimal number of research on the effect of age and gender in kinship verification. Although not investigated directly, different studies highlight the performance gap in verifying kin pairs with the same and different gender.

Introducing the UB Kinship dataset, [10] demonstrates the kinship verification performance is affected by the age gap between the pairs to be verified. Even though the specific age groups of the subjects are not labeled, experiments on the pairs child-old parent and child-young parent result in a verification accuracy difference of about 3.3% in favor of the pairs with a less age gap, revealing the effect of age in kinship verification.

On IIITD database, [6] shows the verification performance gap between the pairs with a different gender. While obtaining an average verification accuracy of 78.5% on parent-child pairs with the same gender, authors report an accuracy of 72.2% on the dissimilar gender parent-child pairs. The same effect is also observed in the comparison of the brother-brother, sister-sister, and brother-sister pairs. On average, verification accuracy of brother-brother and sister-sister pairs is noted as 75.7%, whereas the verification of brother-sister pair is only 68.7%. In the same study, the effect of the age gap between the pairs is also mentioned. Experiments on the UB Kinship database result in around 3% of verification accuracy difference between the pairs child-old parent and child-young parent, indicating the negative effect of age.

In their study, [30] note an average of about 6% and 3% accuracy difference between the same and different gender parent-child pairs on KinFaceW-I and KinFaceW-II, respectively. Also, the verification accuracy on child-young parents surpasses the accuracy in child-old parent pair by 1%.

Dehghan et al. [27] highlight the accuracy difference between the same and different gender pairs as well, reporting a performance difference of about 4.6% and 4.4% on KinFaceW-I and KinFaceW-II, respectively.

[18] reports 5.5% and 2% better accuracies for the same gender parent-child pairs compared to the ones with different gender on KinFaceW-I and KinFaceW-II, respectively. A difference of 4.5% verification accuracy is shown on the UB Kinship dataset, performing better on the pairs with a smaller age gap.

Even though using auxiliary datasets as [9] described in their work, kinship

verification accuracies still considerably suffered from the gender. Although quite similar verification accuracies are recorded on the KinFaceW-II dataset, authors report an accuracy divergence of approximately 3% between the same and different gender parent-child pairs on average on the KinFaceW-I dataset. Similarly, experiments on the WVU dataset support the same effect of gender by yielding 6% difference in favor of similar genders, including the parent-child, brother-brother, brother-sister, and sister-sister pairs all grouped by the gender of pairs. They also noted a 0.5% difference between the child-young parent and child-old parent pairs on the UB Kinship dataset.

1.4 Generative Adversarial Networks

Generative modeling is another hot topic in computer vision, which we use at the core of our study to model the age and gender attributes from face pictures. Since we exploit different approaches from the generative modeling domain, key concepts and specific models necessary to understand our work are described and summarized in this section.

In essence, generative models are trained to learn the distribution of given data in order to generate new samples with similar characteristics to that learned distribution. We can split generative models into two main branches, models that aim to learn an explicit density and models that learn an implicit density. Explicit density modeling requires an explicitly defined density model and solving it to model the given data distribution. PixelRNN, PixelCNN, and variational autoencoders can be given as examples to such models [31, 32, 33]. On the other hand, Implicit density modeling implies that the learned model can sample from a density function without explicitly defining it. To date, generative adversarial networks [34], or GANs, are the most recent and advanced models in this class with their ability to model highly complex distributions such as images. Further details on specific GAN architectures and their characteristics are given in the following subsections of this part. At the same time, more information on explicit density models is not provided since they are out of the scope of our study.

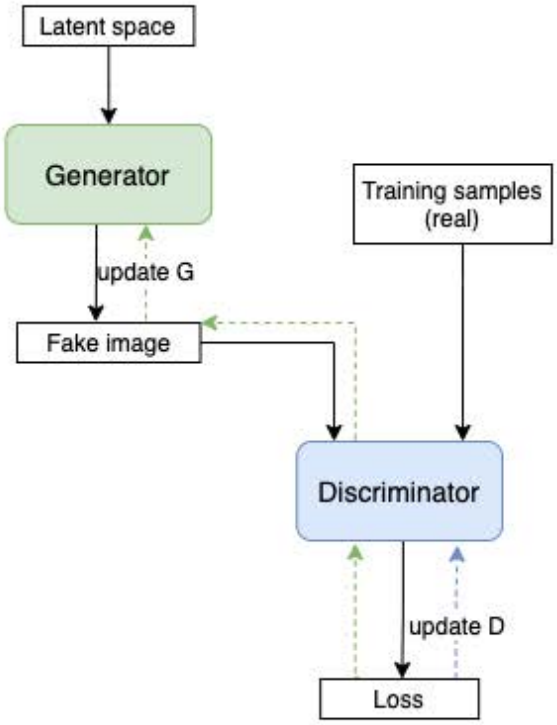


Figure 1.2: Overview of generative adversarial network training

Generative adversarial networks consist of two neural networks that are called generator and discriminator [34]. Training of GANs can be described as a two-player game, in which the generator and discriminator compete against each other as the generator tries to deceive the discriminator by generating real-like samples. In contrast, the discriminator is tasked to distinguish between the real and fake samples. When optimization of the minimax objective function is completed, the generator can generate samples that appear to be real, matching the learned distribution in the training data. An overview of a typical GAN training is shown in Figure 1.2, where generator and discriminator weights are usually updated in turns. Note that while updating the generator weights, discriminator weights are frozen so that they do not receive any gradients. After the training is completed, the discriminator network is removed, and the generator is used to generate realistic samples using its learned latent space.

1.4.1 Manipulation in Latent Space

Latent space in deep learning is indeed the key component behind the learning paradigm. It is a reduced space in terms of dimensionality. The reason deep learning models are trained on any data is to learn new meta representations in that space instead of directly in the image pixel space. That way, if we think of facial representations in the latent space as points, a deep learning model trained on facial images is likely to group faces, say with eyeglasses closer to each other in the latent space, since it learns that this is a common feature. In other words, the model retains the characteristics of the data and simplifies its representation to make it easier to understand.

Training generative adversarial networks on facial images, the model learns the latent space representations of different attributes of faces that are present in the training dataset. Using this simplified form, analysis of the features get easier, and semantically meaningful vector operations can be performed as shown by Radford, Metz, and Chintala [35] in Figure 1.3.

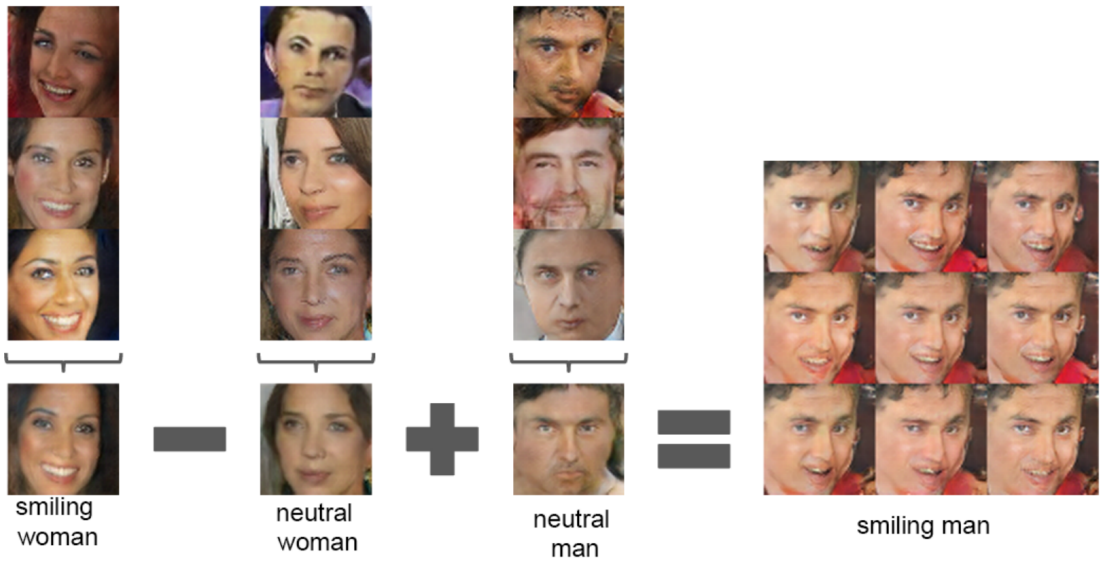


Figure 1.3: Semantically meaningful vector operations in latent space [35]

Shen et al. [36] have conducted a comprehensive analysis on the latent space of pre-trained generative adversarial networks and showed that semantic face editing is possible through vector arithmetic as well as subspace projection without re-training the generator. They prove that a well-trained generator network encodes disentangled semantics in latent space that are usually linearly separable, and when there exists entangled semantics, they can be decoupled by linear subspace projection. Consequently, they validate their hypothesis by performing semantic facial attribute manipulation such as removing eyeglasses or changing the face pose in the latent space, utilizing GAN inversion methods, or encoder-involved models without any extra training.

Although such semantically meaningful operations in latent space can lead to promising applications like manipulating the facial images to add or subtract different attributes arbitrarily, it is not that straightforward. This is because the generative networks are usually incapable of encoding every single feature independently in the latent space due to a number of reasons, such as insufficient data or certain biases in the dataset. That is referred to as entanglement of features in the latent space and is studied in several works [37, 38, 39, 40, 41, 42, 43, 44] to overcome its undesired presence. The entanglement of the features is also visible in the example shown in Figure 1.3. Even though we expect to see a change in only the facial expression of the generated sample, we observe that the background of the sample is changed as well.

We perform manipulations in latent space in order to derive the normalized age and gender attributes in our approach as described in Section 2.1.1.

1.4.2 Style Transfer

Term style transfer in computer vision is used for transforming a source image to exhibit a particular texture style while the original content of the source image is preserved. Style transfer plays a key role in the most recent state-of-the-art generative adversarial networks, as the concept is adapted in the generator architecture to model the different training data attributes.

The first semantically high-level style transfer method is proposed by Gatys, Ecker, and Bethge [45], where a neural algorithm of artistic style is employed for separating and combining the image content with an arbitrary style extracted from another image. To this end, convolutional neural networks are used to extract high-level image information such as content and style of the images, and the style is transferred by minimizing the distance between the generated image's content and style with the target content and style information. This information is encoded in different layers of the convolutional network, and layer-wise operations are performed to compute these distances. An example result from the neural style transfer method is shown in Figure 1.4, content in picture A is preserved, and the arbitrary style is applied to obtain the final image as in B.

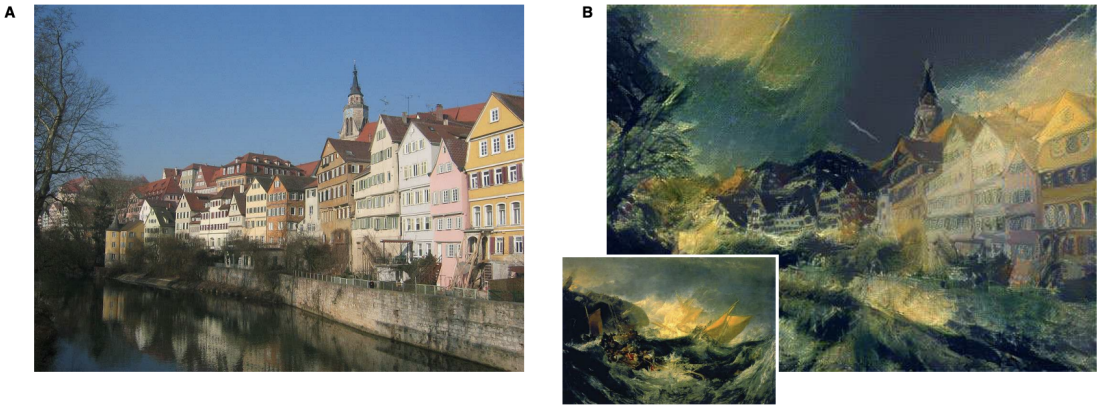


Figure 1.4: A neural style transfer example [45]

In their work, based on the instance normalization [46] technique, Huang and Belongie [47] proposed the adaptive instance normalization, or AdaIN, to perform real-time image style transfer. AdaIN merely adjusts the mean and variance of the content input to match those of the style input given content and style input. Encoding the style and content information as well as the style transfer operation is done in the feature space, which is the first few layers of a fixed VGG-19 [48] network. Recent generative networks employed AdaIN for performing a style-based generation due to the intuition and simplicity behind the AdaIN operation.

More details on these generator architectures are given in the next section.

In our approach described in Section 2.1.2 for age and gender normalization, we treat these two attributes as different styles and try to eliminate their effect while preserving the content of the source image, in our case is the facial identity.

1.4.3 Generator Architectures

This section describes specific generator architectures that we utilize to model the age and gender of facial images.

Figure 1.5: Photo-realistic synthetic face samples [49]

As we strive to perform age and gender normalization on face images, we can define this problem as an image-to-image translation [50]. In this context, we want to transform the face image of a subject into a brand new image in which the age and gender-related attributes are entirely removed or at least suppressed to some extent. Image-to-image translation requires a dataset consisting of matched pairs of images that we want to learn the transformation in between. Creating such datasets is usually costly and sometimes impossible. For instance, if we're

going to model the transformation between male and female genders, we need pairs of images consisting of people with the opposite genders, which is simply impossible.

In their work, Zhu et al. [51] propose a new generative adversarial network CycleGAN, that features a new loss term that is called the cycle consistency loss, which eliminates the necessity of constructing a paired dataset for image-to-image translations. A dataset consisting of images from both domains is sufficient to learn the transformation between both domains instead of having matched opposite gender pairs. In our study, we utilized this architecture to learn a transformation between the male and female domains, which implies modeling both genders.

The generator architecture of CycleGAN follows the architecture in [52], consisting of three fundamental sub-networks that are encoder, transformer, and decoder. The encoder takes in the input image and extracts its features, which are then fed into the transformer network that learns to transform the image to the target domain in the latent space, and finally, the resulting image is synthesized by decoding these transformed features into the image space using the decoder. CycleGAN requires two generators to model the transformation between the two genders, as the transformation from one gender to another is learned by a single generator. A second generator is employed to learn a reverse mapping from the synthesized version to compute the cycle consistency loss. After completing that cycle of transformations from an original image to the opposite gender and then back to the actual gender again, pixel-wise L-1 distance is calculated as the cycle consistency loss term. The intuition behind this is that we should arrive at the same location where we started before performing the transformations. Thus the cycle loss term should be zero ideally.

PatchGANs [53, 54] are used in the discriminator network, which classifies the reduced size patches of 70x70 images as fake or real.

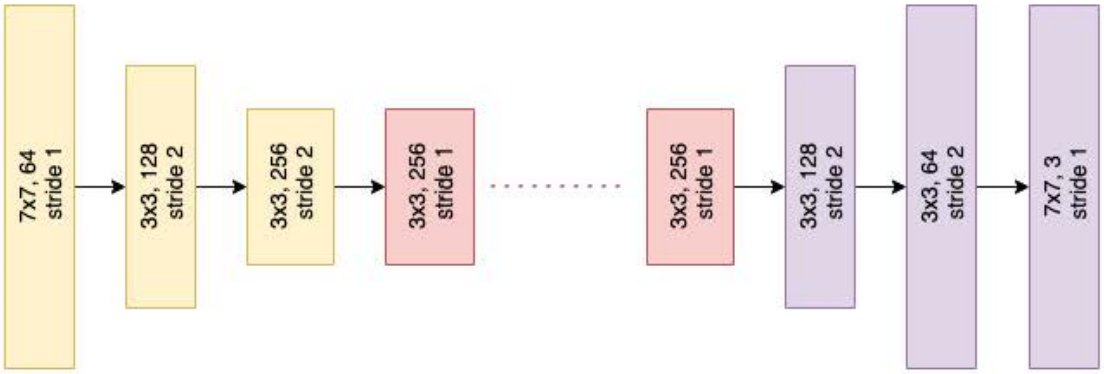


Figure 1.6: CycleGAN generator architecture [51]

Generator architecture of CycleGAN is visualized in Figure 1.6. The blocks represent encoder, transformer, and decoder blocks in yellow, red, and purple colors, respectively. Encoder involves convolutional layers, transformer network consisted of 6 residual blocks [55], and three transposed convolutional layers form decoder.

After the training, we obtain two generators that model the transformations from male to female and vice-versa and make certain manipulations in the latent space of CycleGAN as described in the corresponding section.

Another GAN we utilize in our work is the so-called StyleGAN [49], which introduced the first style-based generator architecture. The authors proposed a novel approach of automatically learning the high-level attributes such as pose or facial identity when trained on face images by encoding these attributes as style vectors. To this end, the generator takes in style vectors at each level of the synthesis blocks from a learned intermediate latent space and normalizes the convolutional layers separately using the AdaIN operation. That way, the generator learns to encode various attributes into each style vector, yielding more powerful modeling of different information present in the training data and finer control over the attributes as the styles are injected at every level of the progressively growing [56] generator layers.

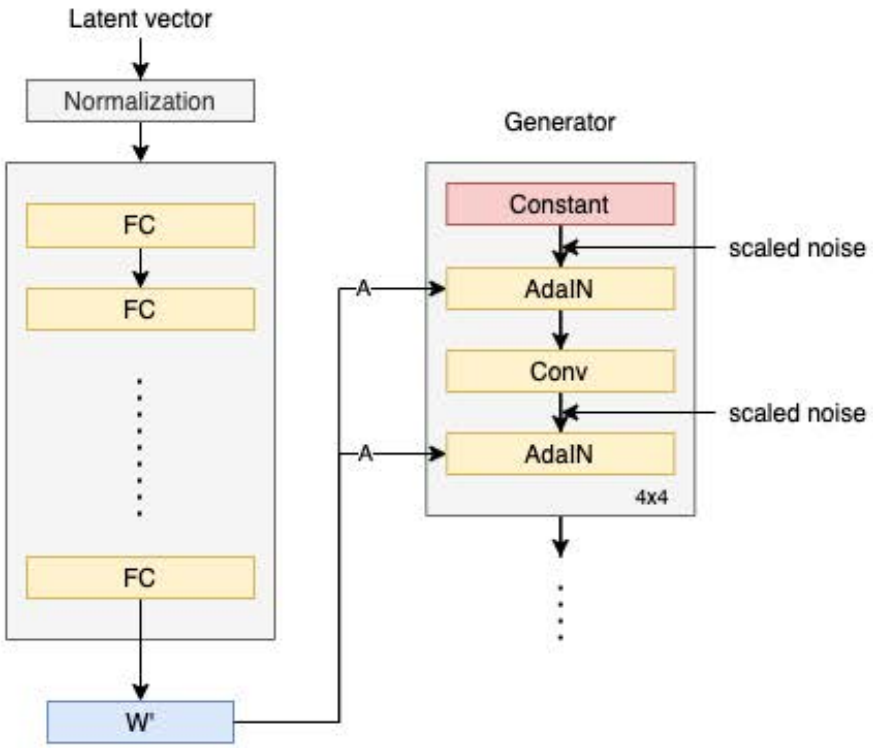


Figure 1.7: StyleGAN generator architecture [49]

Figure 1.7 demonstrates a simplified version of the StyleGAN architecture. Unlike the other generator architectures, the latent vector is not directly fed into the generator but first transformed by a fully connected network, as seen on the left-hand side of the figure. The output of this transformation is the intermediate latent space, which represents a domain-specific manifold. Then this intermediate latent vector W' is fed into each layer of the generator after being through a learned affine transformation that is denoted as A in the figure. Starting from a constant initial image that has dimensions 4×4, the generator progressively upscales it until the desired image size is reached. 2D noise vectors are added into the generation process in order to introduce finer details that the network learns by scaling at each level. Figure 1.5 demonstrates some synthetic face images generated by StyleGAN trained on FFHQ [49] dataset, which look extraordinarily realistic.

Utilizing the powerful modeling capabilities of the StyleGAN, we exploited a pre-trained network on facial images instead of training the network from scratch. Regardless, one has to project an image to be manipulated into StyleGAN's latent space since the generator is not a conditional [57] network. There are two latent spaces as the first latent vector and its transformed version into an intermediate latent space. In our study, we use the intermediate space as our latent space where we manipulated images since it is reported that this space represents more disentangled representation of the face domain [58]. We used StyleGAN to change the age and gender attributes of arbitrary face images by projecting them into the generator's latent space.

Choi et al. [59] proposed another generative adversarial network architecture that is called StarGAN, which basically combines the ideas of [51] and [49]. By employing the cycle loss and a similar generator architecture consisted of an encoder, transformer, and decoder networks as in [51], image-to-image translation is learned with an unpaired dataset of images. Furthermore, style vectors are injected into the upscaling generator layers as in [49], providing a better modeling capacity than CycleGAN even if having a quite similar architecture. Besides, first introduced in [60], StarGAN outstands from the other generative networks with its single generator architecture being capable of modeling multiple domains. Thus, we train a single generator to model different age and gender combinations to analyze the contribution of different age and gender normalization in kinship verification.

In Figure 1.8, StarGAN architecture is visualized, which involves four modules that are generator, mapping network, style encoder, and discriminator. Note that in the figure, there are three domains for illustration purposes. However, it is subject to change in the number of domains in training. During the training of the generator, style vectors are either sampled from a normal distribution or extracted from a reference image. For unconditional style generation, the mapping network takes in a sampled latent vector and outputs the corresponding style vectors to be fed into the generator. For transferring styles from a reference image to the source image, the style encoder network takes in the reference image and outputs its style vectors for all the domains. The discriminator is also a

multi-task network [61, 62], which determines the score of being real or fake corresponding to each of the domains. Note that the architecture of the style encoder and discriminator is the same as visualized in the third column of the figure, the only difference being the networks' output.

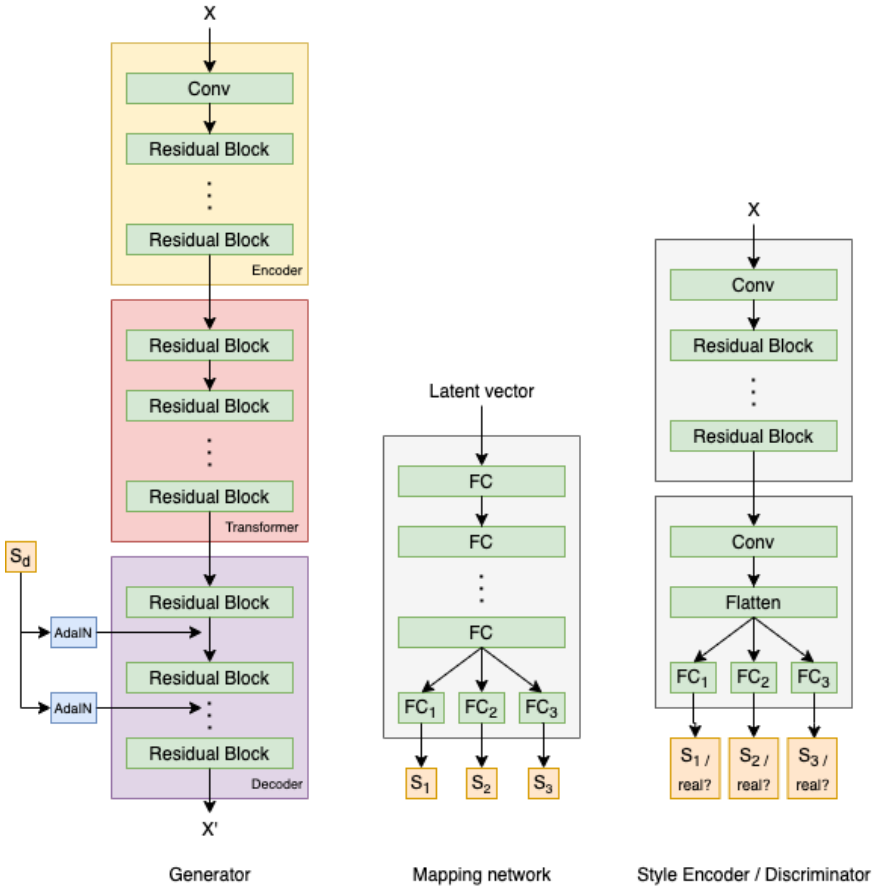


Figure 1.8: StarGAN architecture [59]

We use style encoder and generator networks in our experiments to generate different combinations of age and gender.

1.5 Approach and Contributions

Research in kinship verification has primarily focused on developing specialized deep learning architectures and crafting similarity-oriented metrics. Instead of following a similar path, we define our problem as suppressing the undesired effects of age- and gender-related facial cues regarding kinship verification. We hypothesize that kinship verification performance is degraded as the age gap or the gender differences between the subjects increase, hence, performing kinship verification on a common surface excluding age-and-gender-related features improves the verification performance of an arbitrary kinship model.

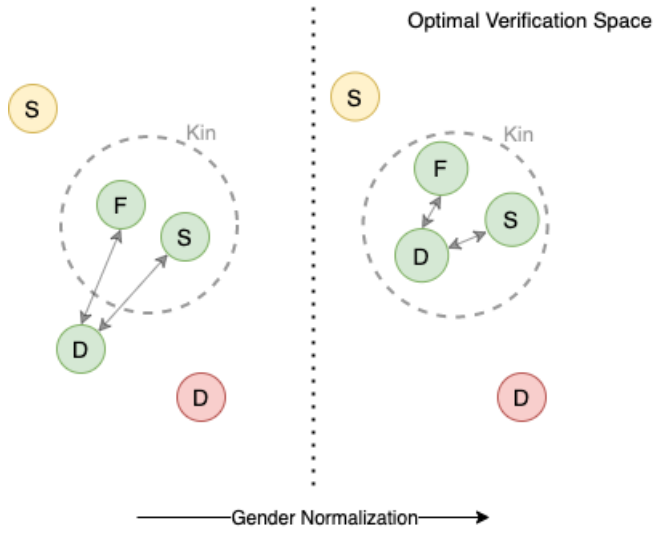


Figure 1.9: Gender normalized latent space in kinship verification

To this end, we model the age and gender attributes of facial images utilizing generative networks and propose several methods to eliminate the undesired effect of these attributes for an improved kinship verification performance. First, we try to find equally spaced latent dimensions by learning the transformation between the two genders, meaning a neutralized gender representation in the generator latent space. Second, inspired by the neural style transfer literature, we consider age and gender as different styles and minimize their effect in facial images

while preserving identity information. Third, we introduce an age- and gender-normalization loss term. By iteratively producing the age-and-gender-normalized version of the input faces as reducing this loss function, we optimize the latent representation of each subject to discard age and gender-specific characteristics. Fourth, we generate faces with different combinations of age and gender and learn a pair-specific weighting of these combinations to model a kinship verification network.

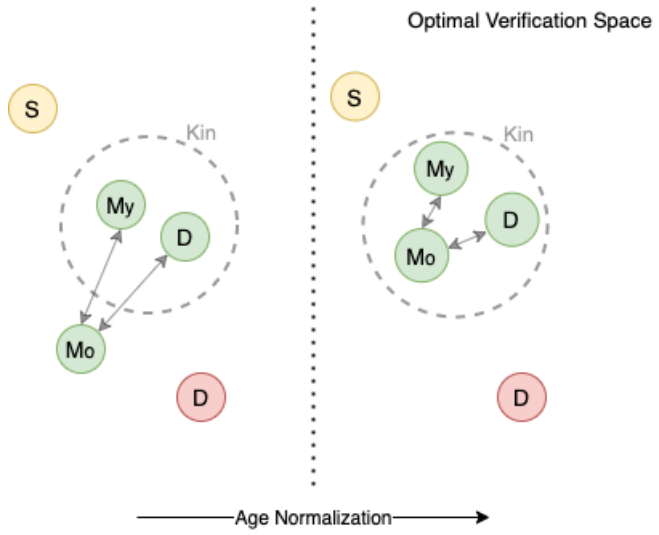


Figure 1.10: Age normalized latent space in kinship verification

Figure 1.9 demonstrates the objective behind normalizing the gender of the subjects to be verified. Abbreviations F, S, and D refer to the relationship types that are father, son, and daughter respectively. The same colors illustrate the ground truth kinship between the subjects, whereas the different colors indicate that there is not a kin relation. Kinship verification usually suffers from gender-inherent facial characteristics, resulting in a deficiency in verification performance, such as the false-negative verification of opposite gender pairs as shown on the left-hand side of the figure. However, suppose we reduce the differences in facial characteristics caused by gender. In that case, we can improve the verification performance of the kinship model, which is shown on the right-hand side of the figure as we transform and solve the verification problem in a more optimal latent

space.

Similarly, Figure 1.10 plots the kinship verification in a standard space versus an age-normalized space. Subscripts y and o refer to the *young* and *old* pictures of the parent subject mother, which is denoted as M. While kinship verification is more difficult when the age gap increases between the parent and child due to the age-related differences in facial characteristics, verification performance is improved in a space where the age-related discrepancy between the facial attributes is reduced. As shown in the figure, younger picture of mother M_y is more easily determined as the parent of her daughter D, whereas the same verification with the older picture M_o of the mother is not correctly performed. When age is normalized, on the other hand, two subjects are accurately verified as kin that is shown on the right-hand side of the figure.

This is the first extensive study investigating the individual and combined impact of age and gender in kinship verification to the best of our knowledge. Besides improving the performance of a recent kinship verification model, we compare the proposed approaches in normalizing the age and gender attributes and conduct detailed experiments on demonstrating the degrading effect of age and gender in kinship verification.

1.6 Thesis Outline

The outline of the rest of the thesis is as follows. Theoretical background and mathematical explanations of all the proposed methods are provided in Chapter 2. Normalizing the age and gender attributes of facial images using generative modeling is described in detail. Chapter 3 defines the used datasets, experimental setup, analysis and comparison of the proposed methods, and detailed discussions on our findings. The thesis is concluded in Chapter 4 with a summary of the contributions, a brief overview of the acquired results, and finally, prospective future paths.

Chapter 2

Kinship Verification Through Age and Gender Normalization

In this chapter, we describe the proposed method for reducing the effect of age and gender-based dissimilarities in the context of kinship verification. Our method consists of three main stages that are the generative modeling of age-and-gender-specific facial attributes, kin relation feature extraction using each of these modeled age and gender combinations, and pair-specific kin relation model weighting followed by the posterior aggregation to determine the kinship score of the input pair.

The design of the overall method is shown in Figure 2.1. G represents the generative modeling block to model different age and gender combinations, K_{n} represent the nth kin relation model for age-and-gender-specific feature extraction, and A_{n} denotes the nth combination's attention module. We show that by reducing the age and gender-related dissimilarities, the proposed method increases the kinship verification accuracy of a recently introduced kinship model on different kinship verification datasets. Our method can be utilized with any kinship model by simply replacing the kin relation models with an arbitrary model. Fundamental components will be further explained and discussed in the following sections of this chapter.

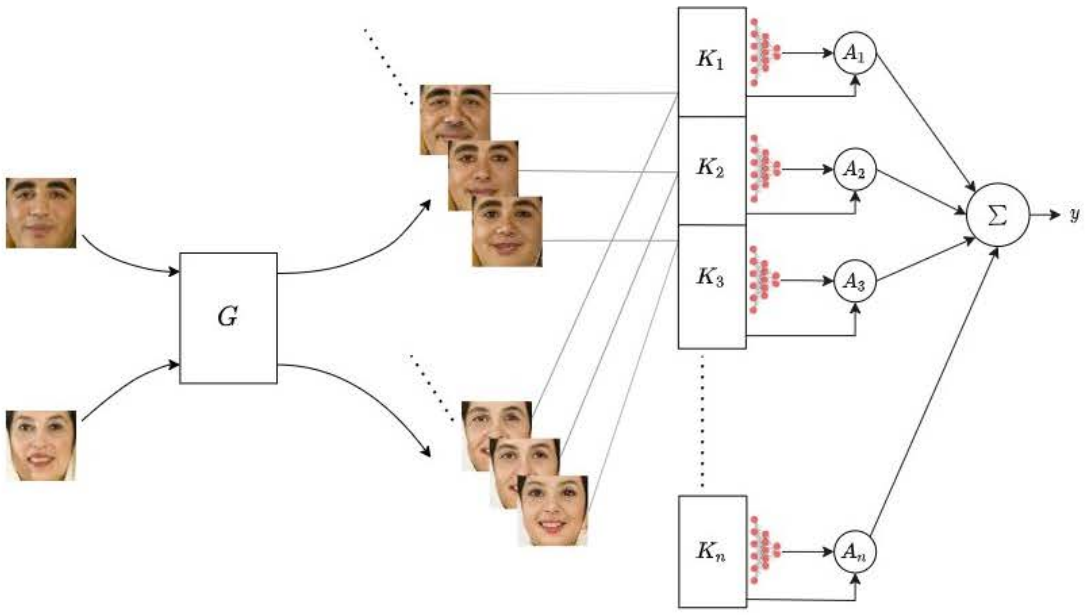


Figure 2.1: Kinship verification with pair-specific age and gender transformation weighting

2.1 Generative Modeling and Normalization of Age and Gender

The first and the core step of our method is the normalization of the age and gender attributes of facial images before performing kinship verification on a pair of images. Normalizing the age or gender attributes of a dataset means that all the samples are transformed into the same medium in terms of these two attributes so that they don't have dissimilarities inherited by age and gender. To this end, we model the age and gender-related attributes using the generative models and normalize them by performing manipulations in both latent and image space. In this phase, a variety of methods can be used for generative modeling. Hence we propose different techniques in terms of tackling the age and gender normalization problem.

In this section, we describe different approaches proposed for age and gender normalization.

2.1.1 Finding Equally Spaced Latent Dimension Between Genders

In order to model an implicit distribution that the data possesses, generative adversarial networks learn an embedding of this distribution that is called a latent space. Therefore, the goal of training GANs is basically to learn this latent space, which represents the distribution of the training data in a lower-dimensional space. Ideally, features that are learned must be independent of each other in latent space, meaning that all the features and their distribution are disentangled.

In Figure 2.2, the latent space of a trained generative adversarial network is demonstrated in its simplest form. The planes in the figure represent the boundaries for each of the learned attributes such as hair type, skin color, or gender. Moving in a perpendicular direction to any of the planes yields a maximum change in the corresponding attribute [36]. For instance, if we traverse the hair type attribute between the curly hair and straight hair domains in the latent space, this should not change the skin color attribute. Ideally, these boundaries must be perpendicular to each other, meaning that the change in one attribute must not affect any additional attribute values.

The number of semantics, or the hyperplanes as visualized in the Figure 2.2, depends on the dataset and how well the generator learned about each of the attributes the dataset contains.

For gender normalization, we suggest that we can neutralize gender by exploiting a pre-trained generator model. In the latent space of the generator, we can project any face to be as close as possible to the hyperplane which separates the gender domains. Then this projection representing the neutralized gender in the latent space is decoded back into the image space.

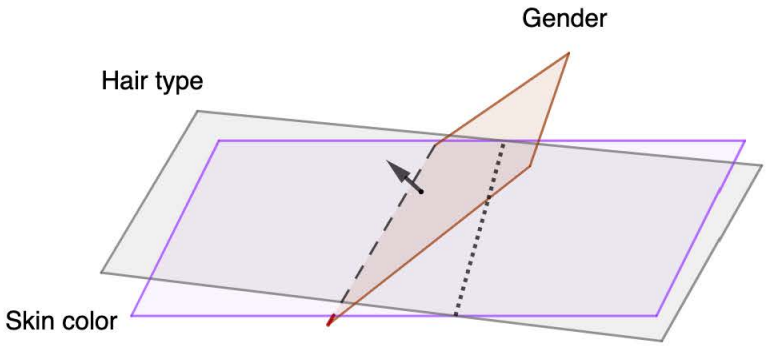


Figure 2.2: Learned latent space consisting of different regions each of which represents an attribute of the modeled data

This objective requires learning a transformation per gender such that $G_1 : I_{\mathrm{M}} \longrightarrow I_{\widetilde{\mathrm{F}}}$ and $G_2 : I_{\mathrm{F}} \longrightarrow I_{\widetilde{\mathrm{M}}}$. Here, $I_{\mathrm{M}} \in \mathcal{R}^3$ and $I_{\mathrm{F}} \in \mathcal{R}^3$ denotes the male and female domains in the image space, whereas $I_{\widetilde{\mathrm{M}}} \in \mathcal{R}^3$ and $I_{\widetilde{\mathrm{F}}} \in \mathcal{R}^3$ represent their synthesized versions respectively. This makes sense since there is not any image representing the neutralized gender so that it is impossible to learn a mapping from I_{M} or I_{F} to the neutral gender directly.

For modeling the afore-mentioned transformations, generator in [51] is employed due to its advantage on training with non-paired data. The generator includes three sub-modules that are the encoder E, transformer T, and decoder D. An input image is first fed into the encoder to be encoded into the latent space, then transformed to the other domain in the latent space, and finally decoded into the image space. Thus, the functions G_1 and G_2 now become

$$\begin{aligned} G_1 &= D_1\Big(T_1\big(E_1(\cdot)\big)\Big) \\ G_2 &= D_2\Big(T_2\big(E_2(\cdot)\big)\Big) \end{aligned} \tag{2.1}$$

where $E_1 : I_{\mathrm{M}} \longrightarrow Z_{\mathrm{M}}$, $T_1 : Z_{\mathrm{M}} \longrightarrow Z_{\widetilde{\mathrm{F}}}$ and $D_1 : Z_{\widetilde{\mathrm{F}}} \longrightarrow I_{\widetilde{\mathrm{F}}}$.

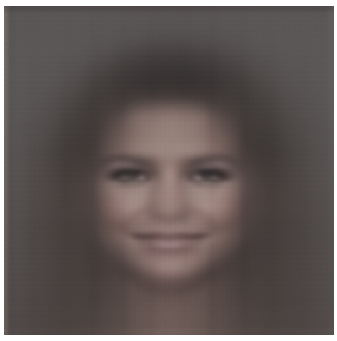
Here, $Z_{\mathrm{M}} \in \mathcal{R}^d$ and $Z_{\mathrm{F}} \in \mathcal{R}^d$ denotes the d-dimensional latent space of the generators for both genders, where $Z_{\widetilde{\mathrm{F}}}$ represents the synthesized female latent vector.

Remember that our purpose is to get gender representations in latent space as the closest possible point to the hyperplane, which defines a boundary between the male and female domains. Since we do not know the formulation of this boundary, we project the neutral gender representation of the dataset onto the line formed by an image and its counter-gender representation in the latent space. Besides, we note that there are no constraints to enforce the linearity of the boundaries, so that we assume the generator is trained well enough to achieve such a latent space.

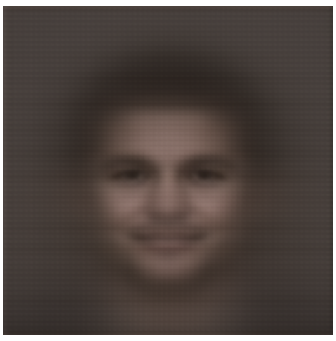
Neutral gender representation Z_μ is computed in the latent space of G_1 and G_2 by encoding all the subjects into latent space and finding the mean face as in the following equation.

$$Z_\mu = \frac{1}{N} \sum^{N} E_1(I_{\mathrm{M}}) + E_2(I_{\mathrm{F}}) \tag{2.2}$$

where N is the number of samples in the dataset.



(a) Meta female



(b) Meta male

Figure 2.3: Meta faces for each gender computed in latent space of the generator

We refer to the generic facial representation as a meta face and derive the neutral gender by using that representation as the fundamental. Meta faces computed in the latent space are visualized by decoding them back into the image space and demonstrated in Figure 2.3.

Then the overall meta face is derived using the faces from both gender domains, yielding a neutral gender representation of the dataset in Figure 2.4

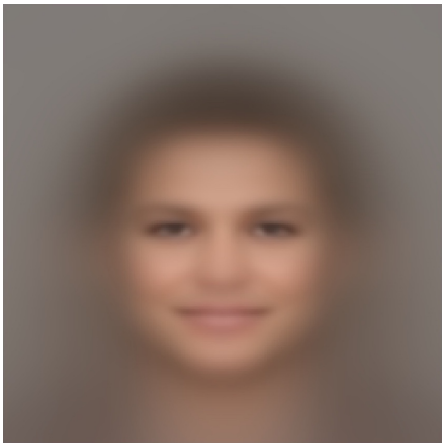


Figure 2.4: Neutral gender representation computed in the latent space

Consequently, normalization of the gender is geometrically interpreted in Figure 2.5 and is described as follows.

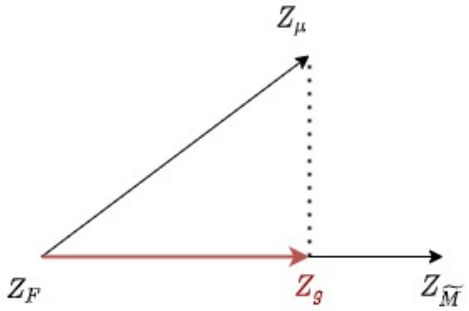


Figure 2.5: Projection of neutral gender onto the line formed by an image and its synthesized opposite-gender representation in the latent space

Z_{F} denotes the original female image and $Z_{\widetilde{\mathrm{M}}}$ is the synthesized male version of Z_{F}. They form a line in the latent space, where the neutral gender Z_{μ} is located somewhere that its projection remains between the opposite gender points. Subsequently, normalized gender Z_g is computed as in Equation 2.3

$$Z_g = \overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}} \cdot \frac{\overrightarrow{Z_{\mu} Z_{\mathrm{F}}} \cdot \overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}}{\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}} \cdot \overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}} \tag{2.3}$$

where

$$\begin{aligned} \overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}} &= T_2\big(E_2(I_{\mathrm{F}})\big) - E_2(I_{\mathrm{F}}) \\ \overrightarrow{Z_{\mu} Z_{\mathrm{F}}} &= Z_{\mu} - E_2(I_{\mathrm{F}}) \end{aligned} \tag{2.4}$$

Finally, Equation 2.3 can be simplified as in the following

$$\begin{aligned} Z_g &= \frac{\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}} \cdot ||\overrightarrow{Z_{\mu} Z_{\mathrm{F}}}|| \cdot ||\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}|| \cdot \cos\theta}{||\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}|| \cdot ||\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}|| \cdot 1} \\ &= \frac{\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}}{||\overrightarrow{Z_{\widetilde{\mathrm{M}}} Z_{\mathrm{F}}}||} \cdot ||\overrightarrow{Z_{\mu} Z_{\mathrm{F}}}|| \cdot \cos\theta \end{aligned} \tag{2.5}$$

Note that if the neutral gender Z_{μ} does not lay between the gender line formed by the both gender end points Z_{F} and $Z_{\widetilde{\mathrm{M}}}$ in latent space, this operation does not work as intended. This is ensured by finding the projected normalized gender vector $\overrightarrow{Z_g Z_{\widetilde{\mathrm{M}}}}$ derived from the synthesized opposite gender $Z_{\widetilde{\mathrm{M}}}$, and comparing its direction with the desired normalized gender vector $\overrightarrow{Z_g Z_{\mathrm{F}}}$.

This comparison is described in Equation 2.6;

$$\frac{\overrightarrow{Z_g Z_{\mathrm{F}}}}{||\overrightarrow{Z_g Z_{\mathrm{F}}}||} \cdot \frac{\overrightarrow{Z_g Z_{\widetilde{\mathrm{M}}}}}{||\overrightarrow{Z_g Z_{\widetilde{\mathrm{M}}}}||} = -1 \tag{2.6}$$

which means that directions of the two opposite middle gender projections are opposite as the geometric interpretation shown in Figure 2.6

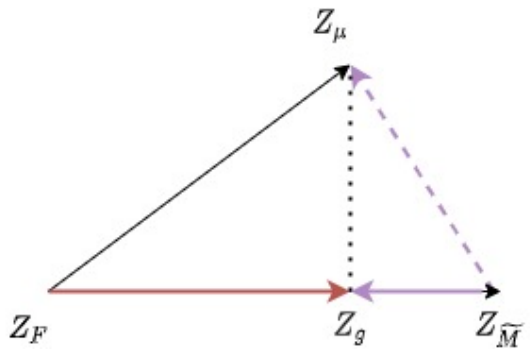


Figure 2.6: Projection of average gender from the female and synthesized male points in latent space

After obtaining the gender normalized representations of I_{M} and I_{F}, we decode them back to image space by

$$I_g = D_{\{1,2\}}(Z_g) \tag{2.7}$$

2.1.2 Identity-Preserved Style Modification for Gender Neutralization

In this section, we propose to normalize gender by extracting the neutral gender style vectors and blending them with the input image style vectors. This is referred to as style-mixing that is first introduced in [49]. Style vectors are powerful representations of the specific attributes learned during GAN training. Therefore

using them to blend different features is an intuitive and convenient way of mixing diverse attributes. The point of using this technique is to preserve identity features of the face image as we use style vector extracted from the original image while transferring non-gender-style extracted from the neutral gender.

To this end, generator architecture in [59] is quite suitable in order to extract the style information of the neutral gender and synthesize them with a given image. This is because it is a conditional generator, which means it takes an image as input and can extract its style vectors that are learned during training using the style encoder sub-module. After the style vectors are extracted from both the input image and the neutral gender, the generator is fed with the combination of these style vectors at specific levels of the upsampling blocks. Style vectors obtained from the identity image help retain the identity information in the synthesized image, whereas the style vectors extracted from the neutral gender are used to suppress gender-specific features that finally result in a normalization of gender.

Let I be the input image and y is the corresponding domain, style extraction can be formulated as

$$\vec{s} = E_y(I) \tag{2.8}$$

where $\vec{s} \in \mathcal{R}^d$ is the extracted d dimensional style vector and E_y is the style encoder network [59]. y denotes the gender-specialized branch of the style encoder, which is trained to learn the male and female distributions in latent space.

So if the identity input and the neutral gender are denoted as $I_{\text{id}} \in \mathcal{R}^3$ and $I_\mu \in \mathcal{R}^3$ respectively, style vectors belonging to both images are

$$\begin{aligned} \vec{s}_{\text{id}} &= E_y(I_{\text{id}}) \\ \vec{s}_\mu &= E_y(I_\mu) \end{aligned} \tag{2.9}$$

Note that I_{id} can either be a male I_{M} or female I_{F} image.

These style vectors are then used to produce a mix of styles combining both identity and neutral gender attributes, which are then used during gender normalized image synthesis. Basically, the style vector of the neutral gender is used at the lower resolution levels of the upsampling blocks of the generator, and the original image styles are injected at the higher levels. This is due to the nature of the training process of the generator, which progressively decodes the latent representation up until the final synthesized image. That way, starting from the transformation block output up until the decoder block output, the generator begins building an image from its latent space by first synthesizing lower levels of resolutions and then increasing the resolution until the final network output. Therefore, at lower resolutions, such as 16×16 images are produced, the rough shape of the face starts to be synthesized, and all the details are formed in the subsequent layers of the decoding block. The overall process is illustrated in Figure 2.7.

Generator illustrated on the left-hand side of the figure can be defined as a function G, consisting of three fundamental blocks that are performing downsampling, transformation and upsampling for image generation. Generator takes in a style vector and an image as input and outputs a synthesized image where the given style is applied as the following equation where $\widetilde{I}$ is the synthesized image;

$$G(I, \vec{s}) = \widetilde{I} \tag{2.10}$$

E represents the encoder network which takes in identity and neutral gender images and output their corresponding style vectors. Before injecting these style vectors into the image generation process, $\vec{s}_\mu$ is projected onto the $\vec{s}_{\text{id}}$ as

$$\vec{s}_g = \frac{\vec{s}_{\text{id}}}{||\vec{s}_{\text{id}}||} \cdot ||\vec{s}_\mu|| \cdot \cos\theta \tag{2.11}$$

where θ is the angle between the vectors $\vec{s}_{\text{id}}$ and $\vec{s}_\mu$ in the latent space and $\vec{s}_g$ is the gender normalized style vector.

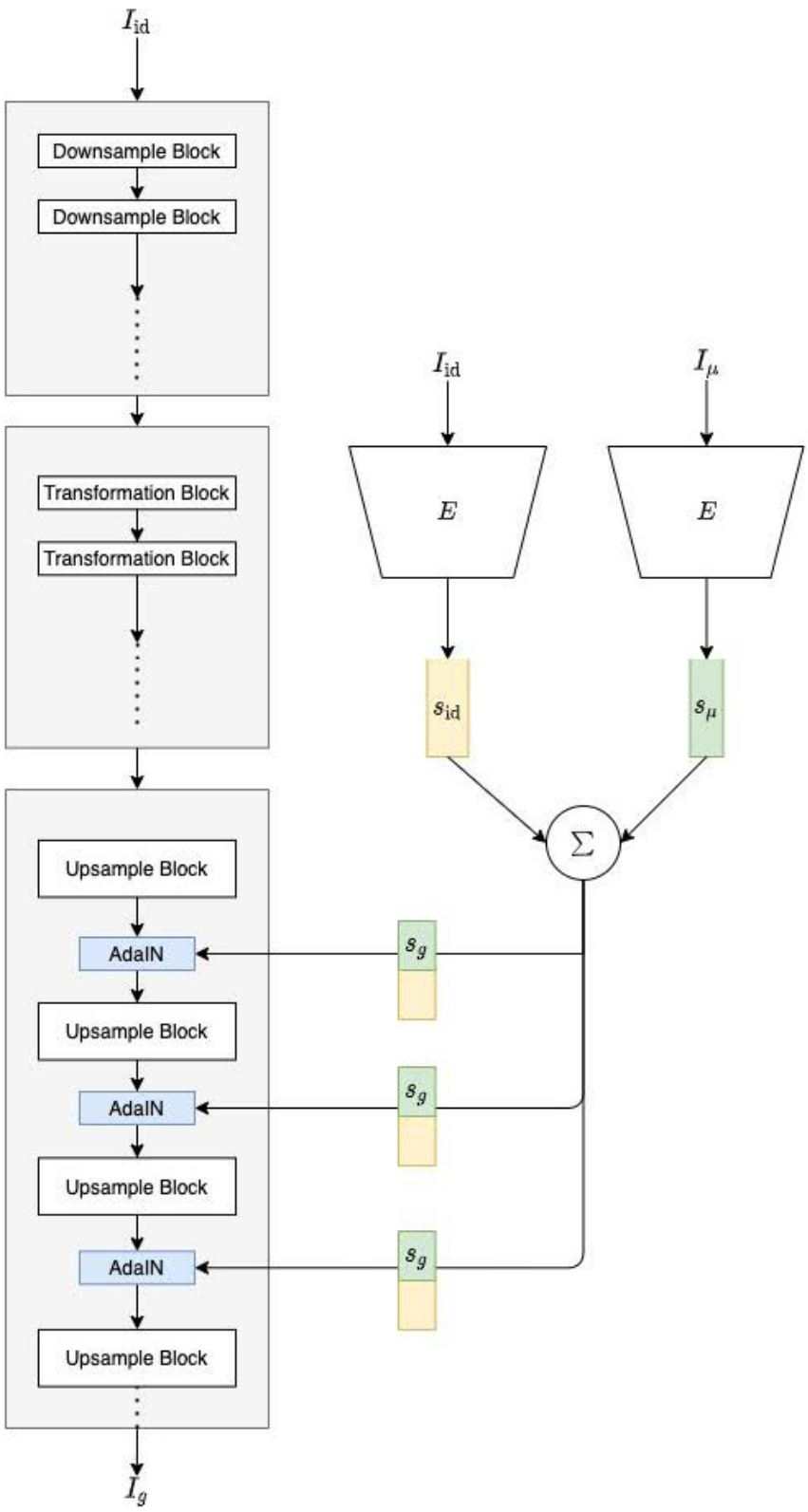


Figure 2.7: Identity-preserved gender normalization by neutral gender style enforcement

Given that, we synthesize the style vectors by injecting the gender normalized style at the lower levels of the decoder block, whereas the identity style vector is injected at the higher levels. That way, the identity-related details remain in the synthesized face, and the neutral gender features are blended without distorting the identity information severely.

Injecting the style vectors to the decoder network is done by the Adaptive Instance Normalization [47] method as proposed in [49]. Basically, if we consider a single resolution level of the decoder or the upsampling block, the style vector of the neutral gender is injected as in Equation 2.12

$$\vec{s}_g \frac{k_n - \mu(k_n)}{\sigma(k_n)} + \vec{s}_g \tag{2.12}$$

where k_n is the nth kernel or the feature map of the decoder block.

Finally, gender normalized style vector $\vec{s}_g$ is injected into G to obtain a gender normalized version of the image, as shown in the following equation.

$$G(I_{\text{id}}, \vec{s}_g) = I_g \tag{2.13}$$

where $I_g \in \mathcal{R}^3$.

2.1.3 Sample-Specific Latent Representation Optimization

All of the prior methods proposed up until here offer a general solution for gender normalization regardless of the input image. In this section, we introduce a sample-based approach towards gender and age normalization, where we optimize the gender and age attributes in an iterative manner by trying to minimize a so-called normalization loss. This requires an intensive search in latent space.

Therefore the generative model that is to be used for this task must be powerful enough in terms of its latent space representation.

We employed the generator architecture in StyleGAN [49], as it has a relatively better latent space residing more disentangled attributes. A disentangled latent space is crucial for performing conditional image synthesis, and [49] utilizes style vectors that are trained to encode different high-level representations of the attributes. In other words, a style vector is expected to encode different information such as the hair type, skin color, facial shape and etc.

Since StyleGAN is an unconditional generator, one cannot simply manipulate an arbitrary image based on a certain condition using this generator. For such manipulation in latent space, the image must be projected to the learned latent space of the generator. So if the image is denoted as I and the generator is G, we need to find G^{-1} such that $G^{-1} : I \longrightarrow Z$ where Z denotes the representation of image I in the latent space. This task is not trivial, therefore [58] suggests increasing the Learned Perceptual Image Patch Similarity [63] abbreviated as LPIPS for the reconstruction of I in latent space as shown in the following equation.

$$\mathcal{L}_{\mathrm{R}} = \mathcal{L}_{\mathrm{P}} + \alpha \sum_{i,j} \mathcal{L}_{i,j} \tag{2.14}$$

where α is a pre-defined hyperparameter with value 10^5 which sets the weight of noise map regularization, $\mathcal{L}_{i,j}$ is the noise map regularization term defined in [58], and $\mathcal{L}_{\mathrm{P}}$ is the perceptual loss that is the LPIPS distance between the target image and the generated image at each iteration.

The perceptual loss basically helps an appropriate reconstruction of the image at each iteration so that a better latent representation of the image is obtained. Perceptual distance is extracted by feeding both the reconstructed and original image into the same pre-trained deep convolutional neural network and computing their corresponding channel-wise distances. Extraction of the channel-wise features is demonstrated in Figure 2.8 where k_n^I is the kernel features extracted from the nth layer for input image I.

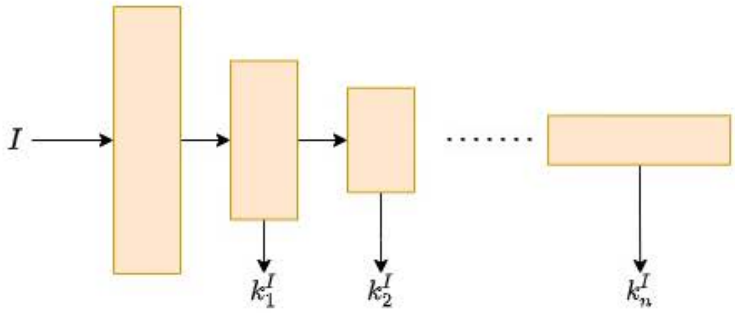


Figure 2.8: Channel-wise features to compute the perceptual distance

Subsequently, if the original image is I_{id} and the reconstructed image is I_{rec}, we can define $\mathcal{L}_{\text{P}}$ between the original and the reconstructed image as in the following equation.

$$\mathcal{L}_{\text{P}} = \sum_{n} ||k_n^{I_{\text{id}}} - k_n^{I_{\text{rec}}}||_2^2 \tag{2.15}$$

Note that the channel normalization is not included in the equation for a simpler definition of the channel-wise distance but is employed as in [63].

Instead of directly reconstructing the face image, we propose to optimize latent representation Z of the image I in such a way that Z yields not only the reconstructed I but also a normalized version of it in terms of age and gender attributes.

To this end, we introduce a new loss $\mathcal{L}_{\text{N}}$, denoting the normalization loss defined in below equation

$$\mathcal{L}_{\text{N}} = \mathcal{L}_{\text{A}} + \mathcal{L}_{\text{G}} + \mathcal{L}_{\text{R}} \tag{2.16}$$

where $\mathcal{L}_R$ denotes the reconstruction loss, $\mathcal{L}_{\text{A}}$ and $\mathcal{L}_{\text{G}}$ denote the age normalization loss and gender normalization loss respectively. Minimizing $\mathcal{L}_{\text{N}}$ yields an age-and-gender-normalized latent space representation of I, while preserving the facial identity information enforced by the reconstruction loss term.

We define gender normalization loss $\mathcal{L}_{\mathrm{G}}$ as in equation

$$\mathcal{L}_{\mathrm{G}} = \lambda_{\mathrm{G}} |0.5 - F_{\mathrm{g}}(x)| \tag{2.17}$$

Here, $F_{\mathrm{g}} : x \longrightarrow P_g$ is an auxiliary network that outputs the gender score P_g of input x at a certain iteration of the normalization phase, λ_{G} is the coefficient of the gender normalization loss. Note that x can either be image I or latent space Z representation of the generated image, depending on whether the gender network takes an image or its latent representation as input in order to determine the neutral gender loss. These two modes of input are demonstrated in Figure 2.9 and Figure 2.10 respectively.

Gender score P_g can take up values between the range $[0, 1]$, representing the network's confidence of a subject being female or male respectively.

Similarly, we define age normalization loss $\mathcal{L}_{\mathrm{A}}$ as in equation

$$\mathcal{L}_{\mathrm{A}} = \lambda_{\mathrm{A}} |a - F_{\mathrm{a}}(x)| \tag{2.18}$$

where a is the value of age that we want to normalize to, $F_{\mathrm{a}} : x \longrightarrow P_a$ is an auxiliary network that outputs the age value P_a of x at a certain iteration of the normalization phase, λ_{A} is the coefficient of the age normalization loss.

We propose minimizing the $\mathcal{L}_N$ using two different approaches. These approaches differ in determining the age and gender score by either using the generated image $G(Z)$ or directly the latent vector Z itself as input. Figures 2.9 and 2.10 summarize both approaches toward the sample-specific optimization process to obtain the gender and age normalized latent representation Z_N of an arbitrary image. Z is initialized as a random latent vector. Then it is updated at each iteration by backpropagation minimizing the normalization loss $\mathcal{L}_N$. G denotes the generator network, while F_{a} and F_{g} represent the auxiliary networks that output

the age and gender score of the Z at a current iteration. F_{p} is a convolutional neural network that is used to extract features in different resolutions demonstrated as in Figure 2.8 to compute the $\mathcal{L}_P$.

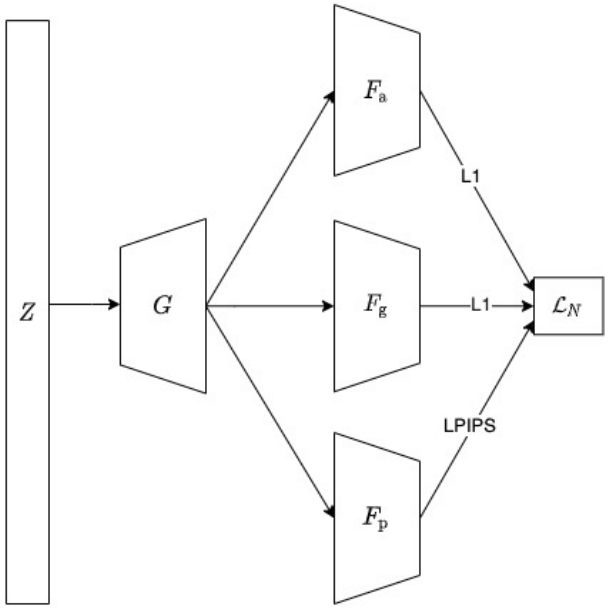


Figure 2.9: Sample-specific latent representation optimization using age and gender feedback in image space

Besides both configurations serve to the same objective of minimizing the $\mathcal{L}_N$, we expect to reduce potential information loss or corruption caused by transforming Z back into the image space by employing F_{a} and F_{g} in a way that yields age and gender scores directly utilizing the latent vector Z as input.

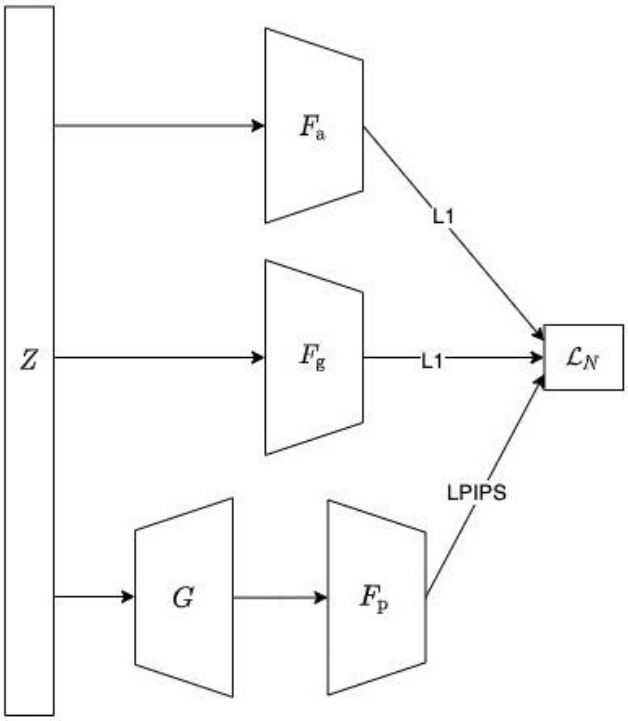


Figure 2.10: Sample-specific latent representation optimization using age and gender feedback in latent space

2.2 Pair-Specific Weighting of Age and Gender Transformations

In this section, we propose an attention-based transformation weighting network that determines the kinship score by utilizing different age and gender combinations of an input image pair. These combinations are also referred to as transformations in the following parts of this section.

This method brings two main benefits in kinship verification. First, instead of utilizing a single age or gender normalized version of the input pair, we exploit a variety of age and gender combinations, yielding a more robust kinship verifier due

to the further reduced dissimilarities in these two attributes. Second, we assign weights to each of these transformations using a dedicated attention module A_n attached to each of the kin models as shown in Figure 2.1. That way, we learn to utilize all of the modeled transformations, assigning them a unique weight indicating the importance of each age and gender combination depending on the input pair.

Generating age and gender combinations is done by synthesizing a range of age and gender versions of both images in the input pair while preserving their identities for each of these synthesized versions. Therefore, given a pair of input images $\{I_1, I_2\}$, G outputs synthesized age and gender versions of each subject such as

$$G\Big(I_1, I_2\Big) = \left\{ \left\{I_1^{d_1}, I_2^{d_1}\right\}, \left\{I_1^{d_2}, I_2^{d_2}\right\}, \ldots, \left\{I_1^{d_n}, I_2^{d_n}\right\} \right\} \tag{2.19}$$

where d_n denotes the nth domain that can be any of the modeled age and gender transformation such as 18 years-old male or 30 years-old female etc.

The number of domains can be increased as many as the number of age and gender combinations synthesized in the generative modeling step. This provides a certain flexibility in the extent of age and gender combinations employed in this step. The more combinations are utilized, the better performance can be expected due to the increased information.

These transformations are then fed into to corresponding kin model K_n, which is specialized for extracting the kin features f_n from a particular transformation. For instance, K_1 might extract the kin features of 18 years-old male version of the input pair, while K_2 extracts the kin features of 30 years-old female version. Therefore, extracting f_n can be formulated as

$$f_n = K_n\Big(I_1^{d_n}, I_2^{d_n}\Big) \tag{2.20}$$

where $f_n \in \mathcal{R}^{64\text{x}13\text{x}13}$.

Kin feature f_n is then used as input to two separate networks that are C_n and A_n, where C_n is the nth transformation's kinship classifier which outputs the probability of being kin y_n for a specific transformation pair, and A_n is the corresponding attention network. Hence the y_n is obtained by

$$y_n = C_n(f_n) \tag{2.21}$$

To obtain the weights w for all transformations, A_n takes f_n as input and outputs a weight score w_n for the corresponding kin score y_n. Then each of these weights are transformed into the probabilistic weight scores, by computing the softmax of all attention network outputs. Finally, all the weight scores for the transformations are obtained as shown in the following equation

$$\begin{aligned} w &= \sigma\big([A_1(f_1),\ A_2(f_2),\ \ldots,\ A_n(f_n)] \big) \\ &= \sigma\big([w_1,\ w_2,\ \ldots,\ w_n] \big) \end{aligned} \tag{2.22}$$

where $w_n \in [0, 1]$ and σ denotes the softmax function that is defined as

$$\sigma(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \tag{2.23}$$

for $i = 1, \ldots, n$ and $x = (w_1, \ldots, w_n) \in \mathcal{R}^n$.

Since we finally have the kinship scores and the corresponding weights for each of the age and gender combinations, we aggregate the weighted kinship probabilities by computing the average posterior and obtain the final kinship score y of input image pair $\{I_1, I_2\}$ as shown in below equation

$$y = \frac{1}{N} \sum_{j=1}^{N} y_j w_j \tag{2.24}$$

where $y \in [0, 1]$ denotes the kinship probability of the input image pair and N is the number of age and gender combinations used to model each of the subjects.

Chapter 3

Experimental Results and Discussion

3.1 Datasets

3.1.1 KFW-I/II: Kin Face in the Wild

To evaluate the effectiveness of the proposed methods in kinship verification, we use the publicly available Kinship Face in the Wild datasets KFW-I and KFW-II [11, 12], which consist of four types of kin relations that are father-son (F-S), father-daughter (F-D), mother-son (M-S), and mother-daughter (M-D). Face images in these datasets are collected without any prior restrictions, meaning that they can differ in terms of lighting, pose, expression, ethnicity, occlusion and so on. All images in both datasets have size of 64×64 containing the cropped and aligned facial images. The fundamental difference between the datasets is that face images in KFW-I are collected from different photos, whereas KFW-II face images are mostly collected from the same photo. Furthermore, KFW-I is an imbalanced dataset in terms of the number of samples in each kin relation, with 156, 134, 116, and 127 number of pairs in F-S, F-D, M-S, M-D respectively.

Number of pairs in KFW-II however is 250 for all the kinship types.

Kin pairs in KFW-I/II are used as positive samples, while random pairs that do not have a kin relation are used as negative samples. These negative samples are randomly constructed for each of the four kin relation subset. For example, a negative M-D pair is formed by replacing the parent or child with another random female parent or child, while the remaining subject is kept the same. Besides, we create a new set of such negative samples for each epoch of training, preventing network to memorize negative samples for any of the subjects thus help generalize better. These shuffled negative samples per epoch are generated once and kept same for all the experiments to maintain a comparable setup between the experiments.

3.1.2 UTKFace: Large Scale Face Dataset

UTKFace dataset [26] consists of over 20.000 face images that are collected in the wild. Along with the non-processed images, dataset provides the post processed versions of the same images that are correspondingly aligned and cropped. Images in the dataset are labeled by their age, gender, and ethnicity, all encoded in the corresponding file names. The dataset consists of 47.7% female and 52.3% male images. Considering the age intervals 0-17, 18-30, and 31-90, there are around 4500, 7200, and 10500 samples, respectively. These labels are generated by using the pretrained [64] network and double checked by a human annotator. Additionally, landmarks of the faces are provided that contain 68 key points.

3.2 Experimental Setup

We utilize UTKFace for mainly modeling the age and gender that are used to normalize these attributes for kinship verification, when we do not use a pre-trained generator network. Separating the training of generative models and the kinship models, we aim to show that pre-trained generative models can easily be

utilized for age and gender normalization based kinship verification. One of the reasons we choose training on UTKFace is that the dataset provides labels for both age and gender attributes of all subjects. The main reason however is that the dataset also includes the non-aligned version of the facial images, which might have been aligned and cropped as we needed. That is required to train a generative model in a way that learning a similar distribution to the aligned images of KFW datasets. Thus, for aligning and cropping the UTKFace images in the same way as the images in KFW, we first detect the mean facial landmark locations utilizing all the KFW-I images. These landmarks are then used to compute a translation and rotation matrices to align the UTKFace images in a similar way to the KFW. Translation is performed by re-positioning the left eye position of each face in UTKFace to match the mean left eye position of KFW dataset. For scaling the faces on the other hand, we compute the distance between left and right eye of the target and perform re-scaling to match the same distance derived from the mean left and right eye positions of KFW. Finally, the aligned facial image is cropped to the size of 64×64 to obtain the full KFW image format.

Concerning the modeling of the age, we pick three age intervals since experimenting with all the specific ages is not practical. These age groups are 0-17, 18-30, and 31-90, taking the effect of aging on facial attributes and the age distribution of the UTK-Face dataset into consideration. The first age group represents the facial characteristics during the pre-adult period of the human, and the second group refers to the next period where the human face changes in a minimal manner, whereas the third group represents the facial attributes during the elderness [65]. There is a relatively higher gap within the last age group because the number of subjects that fall into these three age groups in the UTKFace dataset is also somehow close, yielding relatively fair modeling for the generator during the training.

We use JLNet [66] as our kin relation model and train the network using the exact same parameter setup as proposed without any further hyperparameter tuning, in order to observe the contribution of age and gender normalization throughout different experiments. Note that since we only focus on kinship verification task, we discard the kinship identification head and use only the verification

models trained per kin relation, as in the verification experiments of [66]. Besides, after initializing the network with random weights, we save them to be used as our initial weights for all the experiments to reduce the effect of randomness.

Experiments for all the methods are conducted based on a 5-fold cross validation. We use kinship verification accuracy as our metric to compare the results of all experiments. Same data augmentation techniques are employed for all the experiments, in which we change the contrast, saturation and brightness values of images as well as applying horizontal flip, perspective change and partial cropping all performed randomly in a certain pre-defined range. Although we need to upscale images for age and gender normalization at certain experiments due to utilizing a pre-trained generator network that is trained on higher resolution images, we downscale the generator output back to the dataset's original image size of 64×64 before they are fed into the kinship verification models for consistency between the experiments. Training scheme is kept the same as in [66], where we employed a batch size of 64, used Adam [67] as the parameter optimizer, an initial learning rate of 10^{-4} with a step decay of step size 40. Finally, weighted cross entropy is used as the network loss function for each verification output, setting the weights to $[0.25, 8]$ for the negative and positive samples respectively. As the only difference, we trained the model for 100 epochs which is fewer than [66] to prevent overfitting since we did not need to train the identification module.

Considering some of the experiments required auxiliary networks and extra procedure for modeling the age and gender, further details regarding to these specific experiments are provided in the corresponding subsections of the chapter.

3.3 Gender Normalization in Kinship Verification

In this section, we evaluate the contribution of gender normalization in kinship verification, conducting an analysis using three different approaches to assess the

effect of gender. Our first approach is to normalize genders of all the subjects in the dataset, meaning that all of the genders are represented in a neutralized manner rather than being male or female. To further support the idea that gender-related dissimilarities degrade kinship verification performance, we conduct two additional experiments in the context of gender normalization, by transforming all the subjects to one of the genders, male or female. Hence, we assert the idea that gender has an impact on kinship verification and we can reduce its effect by simply representing all the samples on a common surface even though we do not normalize gender of all the subjects.

In order to normalize gender in the image space, we employ the methods described in Sections 2.1.1, 2.1.2, and 2.1.3. Concerning the sample-specific normalization method, since the generator is trained on a spatially aligned facial images with a size of 256×256, we first align the KFW images at the same location concerning the facial key points and upscale them to be compatible with the pre-trained StyleGAN architecture. This process requires facial landmark detection, so we employ [68] for the landmark detection and align faces to the same location as in [49]. In practice, landmark detection algorithm cannot detect all the faces, especially for the face images that are too small such as KFW. Thus, although we are able to align majority of the faces in the dataset, some of them are not appropriately aligned due to an imperfect landmark detection. These non-aligned images are not used for the gender normalization due to their divergent distribution that mismatches the distribution modeled by the pre-trained generator. To avoid experimenting with the lacking number of data, we still include these non normalized subjects in our training. To this end, we ensure that any gender normalized subject would not match its non-normalized pair, so that replace the normalized subject with its original version which has its pair failed in the alignment step. This prevents the experiment being contaminated while help use of all the samples in the dataset.

For implementing the gender network F_{g}, we performed transfer learning with a pre-trained facial recognition network called SE-ResNet [69], that is trained on VGGFace2 [70]. Using SE-ResNet as the feature extractor we trained a gender classifier on CelebA-HQ [56] dataset. Note that F_{a} is not required for this part

of the experiments.

By using the gender classifier as a feedback in the gender normalization loss term as employed Equation 2.17, gender score converges to a normalized value that is 0.5, since the gender labels can take 0 or 1 for female and male respectively.

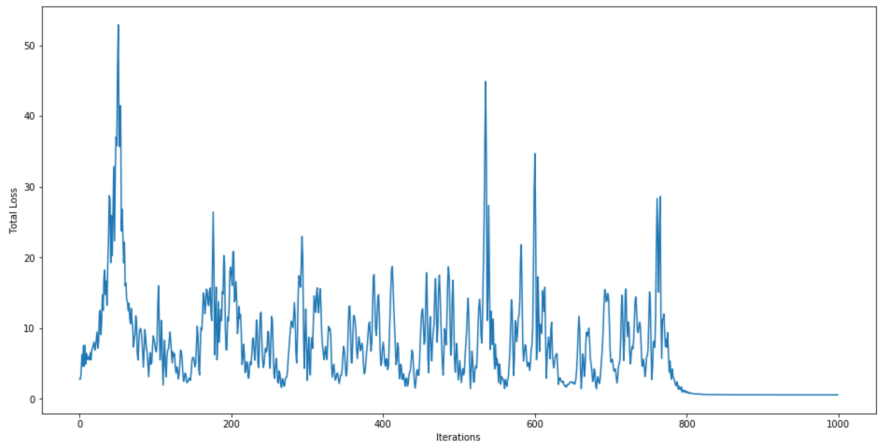


Figure 3.1: Overall normalization loss during the gender normalization process for an example subject

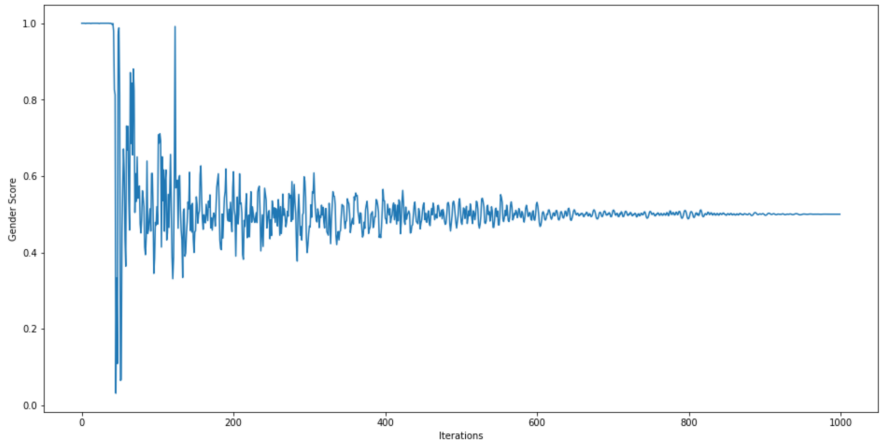


Figure 3.2: Gender score inferred by F_g during the gender normalization process for an example subject

Figures 3.1 and 3.2 shows the value of overall loss and the corresponding gender

score for each iteration of the gender normalization for a single subject, demonstrated for a better comprehension of the process. In essence, gender normalized reconstruction of the subject's latent representation proceeds with high oscillations as seen in Figure 3.1 and the gender score converges to 0.5 until the end of the process. Notice that, this is the procedure for normalizing the gender to obtain the neutralized gender version of the subjects. As explained previously, other two approaches concerning the analysis of gender involve transforming the gender of child or parent for F-D and M-S kin pairs. To this end, instead of normalizing the gender scores of the subjects to 0.5, we converge them to either 0 or 1.

Concerning the neutral gender representation needed for the methods in Sections 2.1.1 and 2.1.2, we utilize the meta face computed from over 60k images in the latent space of [51]. That representation is utilized in the projection of the neutral gender to find the equally distanced gender point in the latent space as described in Section 2.1.1. Besides, the same representation is used in extracting and normalizing the neutral gender style vector as explained in Section 2.1.2.

Table 3.1 shows the results of different gender-focused experiments described so far. *Baseline* indicates the kinship verification experiment results published in [66] that is the model trained without gender normalization. *Latent Projection* shows the results for the same network trained with employing the gender normalization method of finding the equally spaced latent dimensions, *Style Modification* is the method for gender normalization by the style vector modification, and finally *Sample-Specific* shows the results of which the sample-specific gender normalization method is utilized. Rows *Child Gender* and *Parent Gender* on the other hand indicate the results of gender transformation in only F-D and M-S pairs. *Child Gender* means the parents' gender in the F-D and M-S pairs are transformed into the children's gender. By contrast, *Parent Gender* means the children's gender in the those pairs are transformed into the parents' gender.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Baseline	0.6608	0.7309	0.7207	0.5897	0.6755	0.6800	0.7140	0.6860	0.7060	0.6965
Latent Projection	0.6612	0.7217	0.7123	0.5905	0.6714	0.6850	0.7100	0.6820	0.7080	0.6962
Style Modification	0.6510	0.7014	0.7026	0.5807	0.6589	0.6750	0.7080	0.6780	0.7010	0.6905
Sample-Specific	0.7093	0.7212	0.7481	0.6451	0.7181	0.7720	0.8000	0.8030	0.7880	0.7907
Child Gender	0.7426	0.7698	0.7404	0.6853	0.7345	0.7920	0.7650	0.7570	0.7740	0.7720
Parent Gender	0.7393	0.7635	0.7287	0.6674	0.7247	0.7900	0.7670	0.7580	0.7840	0.7747

Table 3.1: Gender normalization in kinship verification analysis. *Latent Projection* shows the results kinship verification results of the network that is trained as employing the gender normalization method of finding the equally spaced latent dimensions, *Style Modification* is the method for gender normalization by the style vector modification, and *Sample-Specific* shows the results of which the sample-specific gender normalization method is utilized.

Experiment results show that except the first two methods, gender normalization has improved the kinship verification accuracy on both datasets KFW-I/II. As for the kinship verification mean accuracy, sample-specific gender normalization increase the accuracy about 4.3% for KFW-I and about 9.5% for KFW-II compared to the baseline. We do not obtain a similar improvement using the first two methods, even resulting in a slight performance decrease when we employ the style vector modification method. Besides not improving the opposite gender pair verification, these two methods cause varying performance drop in the pairs with the similar gender. This can be explained as when these methods are applied, they somehow distort the kinship-oriented features rather than normalizing the gender-only characteristics of the faces. When we further investigate the effectiveness of these two methods, we observe that the first method where we find an equally spaced latent dimensions modifies the facial images in a quite limited extent. Second method where we applied style modification on the other hand, leads to a more notable changes in the facial images, in which the identity characteristics are effected in a varying magnitude. Thus, the experiment results show

a similar pattern in which the first normalization method does not notably effect the verification performance while the second method degrades the performance to some extent.

Similar improvement to sample-specific gender normalization is also observed for the verification performance where we converged parent or child gender to their opposite for F-D and M-S pairs. For the case where parent gender is converged to child gender, there is a 6% and 7.5% accuracy improvements on the corresponding datasets. In parallel, for the case where child gender converged to parent gender there are improvements of 5% and 7.8% for KFW-I and KFW-II respectively.

On the other hand, if we take a detailed look at the pair-wise scores, there is a certain verification performance gap for the baseline in favour of the pairs with same gender compared to the pairs with opposite gender on both datasets, especially on KFW-I. This difference notably decreases when we employ gender normalization or reduces its effect by converging to a single gender. For F-D pair, gender normalization based verification increases accuracy from about 66% to 71% on KFW-I, while the same pair accuracy raises from 68% to 77% on KFW-II. Performance on M-S pair shows a similar improvement, enhancing the baseline score of 59% to 64.5% on KFW-I and 70.6% to 78.8% on KFW-II. The magnitude of improvement naturally differs between the datasets, since there are higher number of samples present in KFW-II, hence the influence of normalization is clearer.

Table 3.2 highlights this improvement by providing more insight on similar and dissimilar gender pair verification accuracies and their difference for each of the experiments. Concerning the results of baseline on KFW-I, similar gender kin pairs F-S and M-D has an average accuracy of 72.6% while F-D and M-S has an accuracy of 62.5%. By contrast, these accuracies are 73.5% and 67.7% correspondingly for the sample-specific gender normalized verification. All in all, divergence in the verification of different kin pairs is reduced from 10% to $5-4\%$ interval for all the gender experiments, indicating that the verification deficiency due to gender dissimilarities is considerably diminished. Although the baseline performs quite the same on KFW-II by means of similar and dissimilar gender

pairs, we still observe improvements in verification with dissimilar genders as they even surpass the performance of pairs with similar gender. Note that the negative values in table indicate that the verification performance of dissimilar gender pairs are better than the performance of the pairs with same gender. Lastly, although the performance difference between the similar and dissimilar gender pairs seems to be decreased for the first two gender normalization methods, it should be noted that the overall performance is dropped and thus this is not a desirable case.

	KFW-I			KFW-II		
	μ_s	μ_d	$\mu_s - \mu_d$	μ_s	μ_d	$\mu_s - \mu_d$
Baseline	0.7258	0.6252	0.1006	0.7000	0.6930	0.0070
Latent Projection	0.7170	0.6258	0.0912	0.6960	0.6965	−0.0005
Style Modification	0.7020	0.6158	0.0862	0.6930	0.6880	0.0050
Sample-Specific	0.7346	0.6772	0.0574	0.8015	0.7800	0.0215
Child Gender	0.7551	0.7140	0.0411	0.7610	0.7830	−0.0220
Parent Gender	0.7461	0.7033	0.0428	0.7625	0.7870	−0.0245

Table 3.2: Gender normalization effect on similar and dissimilar gender kin pairs. μ_s denotes the the mean accuracy of same gender kin pairs F-S and M-D, whereas μ_d is the mean accuracy of different gender kin pairs F-D and M-S. $\mu_s - \mu_d$ represents the accuracy difference between the μ_s and μ_d

For the following parts of the experiments, we employ only the best performing gender normalization which is the sample-specific normalization method, unless otherwise is stated.

3.4 Age Normalization in Kinship Verification

In this part of the experiments, we analyze the effect of age in kinship verification. The main purpose here is to show that if we remove the age-related dissimilarities in human face by preserving the identity, we can learn a better kin model which is robust to detect kinship probability of pairs with photos in different ages. To this end, we normalize the age of all the subjects, by transforming them to three different age groups that are 18, 35 and 55 years old. These ages are selected due to their expressiveness in terms of different periods of a human face during its aging. For this experiment, we do not leave any subject out of the normalization like we did in the parent and child gender experiments since this is not required in case of analyzing the effect of age. But remember that non-aligned faces cannot be used for the age normalization as well, due to avoiding a distribution mismatch with the generative model. Lastly, we also experiment with an independent age transformer network introduced in [71], in order to compare with the age normalization method we propose.

Similar to the gender normalization, we employ the method that is demonstrated in Figure 2.9 for age normalization, with removing the F_{g} network for analyzing the age-only affect on verification. Since we use the same generator, all the images are re-scaled and aligned before the age normalization process as in Section 3.3.

For implementing the network F_{a}, we use [69] again as our backbone and train an age regression network by means of transfer learning from facial recognition task to age prediction. Hence, F_{a} consist of a 2-layer fully connected network on top of [69] that is employed as the facial feature extractor. The network is trained on UTKFace [26] dataset which consist of over 20.000 facial images with ages varying from 0 to 116 years old. We up-scale images to size 224×224 to be consistent with the generator which would be required for the normalization phase. During training of F_{a}, we apply online data augmentation by random horizontal flip and random crop. Concerning the training setup, we freeze the pre-trained [69] except the last convolutional block, which is fine-tuned by a

learning rate of 0.00001 for extracting the age-related features. For the fully connected layers attached for regression, however, we employ a higher learning rate of 0.0001 since they are trained from scratch unlike the feature extractor. We use a batch size of 64, maximum number of epochs 50 bound to early stopping with patience of 6 epochs. Learning rate decay on plateau with a decay rate of 2 is applied to prevent learning from stagnating. L1 distance is used as the loss metric for training the age regression network.

Using the trained F_{a} and minimizing the age normalization loss in Equation 2.18, the age of the subject is normalized by converging to the desired age value for each of the experiments. For instance, if we transform all the subjects to their 18 years-old version, parameter a in Equation 2.18 is assigned the value 18. For a better comprehension of the age normalization process, the total normalization loss, L1 distance to the desired age value, and the age value at each iteration are showed for a single subject in the following figures.

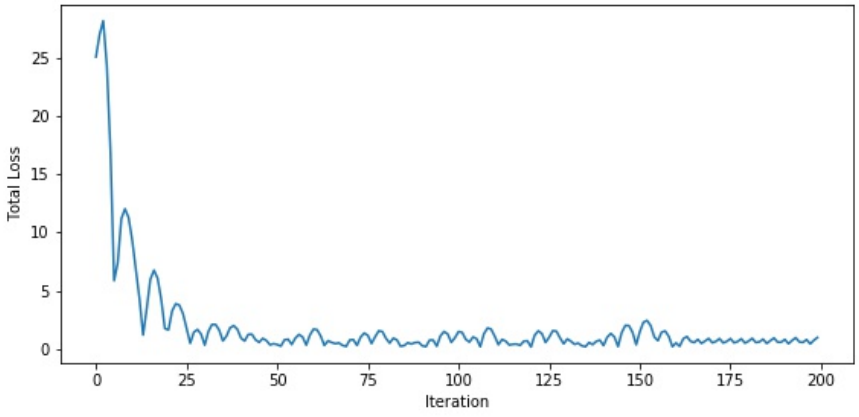


Figure 3.3: Total normalization loss for an example subject during the age normalization process

Demonstrated in the figure above, overall normalization loss is computed according to the Equation 2.16, excluding the gender normalization term for the sake of the age-normalization-only experiments. The normalization loss decreases and finds a local minima as we continue to iterate up to a certain point. Note

that we held the number of iterations smaller than the case where we normalized the gender of a subject, since it converges faster for the age normalization.

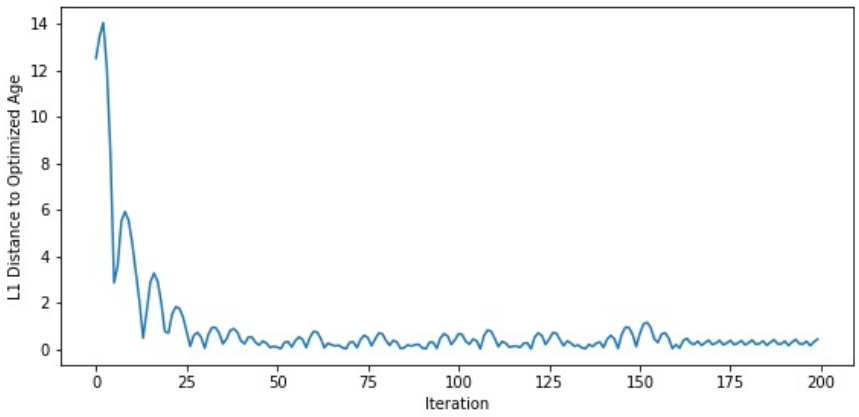


Figure 3.4: Distance to the target age, in this case 18 years old, computed by subtracting the current age value inferred by F_{a} from the target age during the age normalization process for an example subject

L1 distance to the target age for the same specific subject also decreases in parallel with the overall loss, as long as the normalization process continues. For this case, subject is 13 years older than the target age at the beginning, which is then normalized to have a minimal distance to the normalization target so that the process is finished with a distance of between $0-1$ years to the target.

Age value measured at each step of the normalization process shown in figure above also recaps that the subject is predicted as 30 years old before starting to the normalization and then gradually converges to the target age, which is 18 years old, until the end of the process.

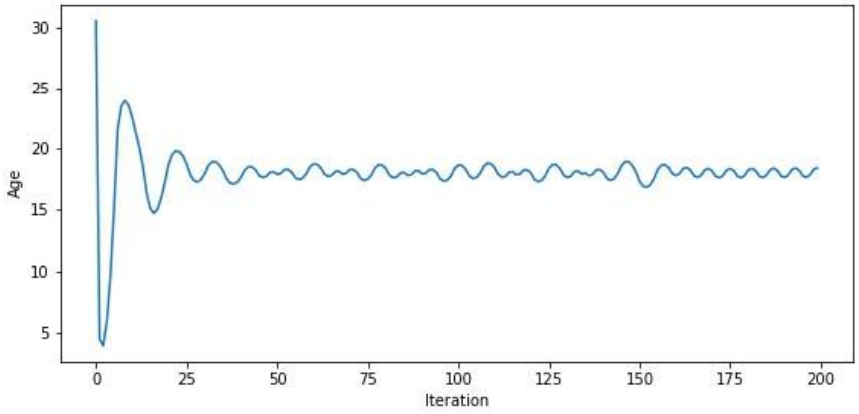


Figure 3.5: Age value per iteration inferred by F_{a} during the age normalization process for an example subject

Table 3.3 shows the results of age normalization based kinship verification using the verification accuracy as the evaluation metric. *Group 1* denotes the experiment in which the age of all subjects are normalized to 18 years old, whereas *Group 2* and *Group 3* represents the age normalization at 35 and 55 years old respectively. We also conduct a separate age normalization experiment using a pre-trained age transformation network that is recently introduced in [71] without making any modifications for the comparison purposes. Since the network is trained on different range of ages instead of exact age values, we transform all the subjects to the age interval of 15-17 using this network. The result of this experiments can be found in the last row of the table.

Experiment results show that age normalization improves the kinship verification accuracy on both datasets KFW-I/II compared to the baseline performance. Considering the overall performance on kinship verification, normalization at all three age groups improve the accuracy by at least 4.2% for KFW-I, and 9% for KFW-II.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Baseline	0.6608	0.7309	0.7207	0.5897	0.6755	0.6800	0.7140	0.6860	0.7060	0.6965
Group 1	0.7167	0.7083	0.7519	0.6940	0.7177	0.7840	0.7960	0.8140	0.7900	0.7960
Group 2	0.7241	0.7149	0.7481	0.7027	0.7225	0.7880	0.7880	0.7980	0.7700	0.7860
Group 3	0.7207	0.718	0.7559	0.6940	0.7221	0.7760	0.7960	0.8080	0.7820	0.7905
Or-El *et al.* [71]	0.6272	0.7120	0.7161	0.5891	0.6611	—	—	—	—	—

Table 3.3: Age normalization in kinship verification analysis

Again, the magnitude of improvement between the two datasets differ because of different number of subjects present, the dataset with more samples exhibiting the normalization effect distinctly. While normalization at age group 2 performs slightly better than the remaining age groups for KFW-I, normalization at age group 1 outperforms the rest for the KFW-II dataset. Possible reasons for that are mentioned in the following paragraphs. In general, the improvements in age normalization seems within the ±1% range in comparison to the improvements in gender normalization that are reported in Table 3.1.

Focusing on the pair-wise verification scores at KFW-I, normalization at age group 2 outperforms the rest for F-D and M-S kin pairs, whereas the best performance for F-S and M-D pairs obtained by the age group 3 normalization. The most significant improvement compared to the baseline is in M-S kinship verification, as the accuracy is increased from 59% to about 70%. Nevertheless, improvements between the normalization at different age groups differ slightly so that yielding a rather consistent enhancement trend. For the experiments on KFW-II on the other hand, we observe that normalization at age group 1 leads a better verification performance for M-D and M-S pairs, whereas there is not any group that performs superior than the normalization at the other ages for pairs F-D and M-S.

Improvements between different age groups vary due to a number of reasons

including the mismatching age distribution between the datasets and the kin pairs. Although there is not ground truth information for KFW datasets by means of subject ages, the higher performance boost in M-S pair verification might imply a more divergent age profile of the mother and son pictures compared to the other kin pairs. Besides, the quality of normalization at different age groups highly depends on the distribution learned by the generative model, which directly effects the kinship verification performances reported.

Lastly, the age normalization performed by the pre-trained [71] results in the worst verification performance in our experiments. We observe that for KFW datasets, the age transformation by the network distorts pretty much the identity characteristics of faces as shown in the appendix of this thesis, even if we align images to match the learned distribution before they are fed into the network. Since identity is more-or-less corrupted in such a manner, performing the verification worse than baseline is indeed a natural consequence.

3.5 Age and Gender Normalization in Kinship Verification

We observed that the age and gender normalization enhance the kinship verification performance by analyzing their sole effect in the previous sections. Therefore, in this section we investigate their combined efficacy in verification. In addition, we study the impact of age and gender normalization in latent space, unlike the previous experiments that are realized only in image space. Our intention is to reduce the loss of information in various steps caused by the additional reconstructions, such as networks F_{a} and F_{g} predicting the age and gender scores in the reconstructed images at every iteration. Here reconstruction means that a latent vector being transformed to the image space by the generator network. An example to this can be seen in Figure 2.9, whereby the generator reconstructs the updated latent vector Z as an image which is then fed into the networks F_{a} and F_{g} for age and gender prediction. To this end, as well as analyzing the combined

effect of age and gender normalization in kinship verification, we also investigate the potential performance gain by evaluating the age and gender directly in latent space by changing networks F_{a} and F_{g} such that they can directly take the latent vector as their input. Note that we employ only the best performing approaches in the following experiments for the sake of avoiding redundant complexity in the analysis.

3.5.1 Normalization in Image Space

In this part of the experiments, we analyze the combined effect of age and gender normalization on kinship verification in image space using the methods that are explained in sections 2.1.3 and 2.2, implemented using the generative model architectures [58] and [59] respectively.

To investigate the simultaneous normalization of age and gender as proposed in Section 2.1.3, we normalize the subjects to the same three different age groups in Section 3.4 as well as to the neutral gender. The images are normalized using the complete loss term in Equation 2.16 unlike the previous experiments in which we employed only the age or gender loss terms within the normalization loss. The networks F_{a} and F_{g} are held the same with the previous experiments and used together as shown in Figure 2.10 to compute the age and gender losses.

Experiment results in Table 3.4 show that the simultaneous normalization of the age and gender improves the kinship verification accuracy on both datasets compared to the baseline. Besides, we observe that the simultaneous normalization of the age and gender performs slightly worse than the experiments where we normalize the age and gender on KFW-I. However, this is not the case for KFW-II, as the gender normalized versions of the age groups 2 and 3 yield a better verification performance. So we may infer that for KFW-I, compared to the normalization of only one attribute, normalizing the age and gender simultaneously might have effected the facial identity rather than yielding a beneficial normalization that is supposed to improve the verification performance. For KFW-II on the other hand, simultaneous normalization of age and gender results in an

improvement on verification although it is as minor as only about 1%.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Baseline	0.6608	0.7309	0.7207	0.5897	0.6755	0.6800	0.7140	0.6860	0.7060	0.6965
Gender-Neutral Group1	0.7054	0.7115	0.7327	0.6815	0.7060	0.7780	0.7900	0.7900	0.7860	0.7860
Gender-Neutral Group2	0.7019	0.7244	0.7601	0.6850	0.7178	0.7920	0.8060	0.8100	0.7800	0.7970
Gender-Neutral Group3	0.7170	0.7308	0.7319	0.6850	0.7162	0.7840	0.7880	0.8100	0.7860	0.7920

Table 3.4: Sample-specific normalization of age and gender simultaneously

In the following set of experiments, we analyze the benefit of pair-specific weighting of different age and gender combinations that is proposed in Section 2.2. As a reminder, unlike the previous methods where we normalized the age and gender to have a common value for all the subjects, in this method we utilize different combinations of age and gender attributes of input pairs for the kinship verification. For modeling these combinations, we train a StarGAN [59] model with certain modifications. The motivation of utilizing this model was that training a single generator is sufficient to model multiple domains, such as age and gender, unlike the other architectures requiring a dedicated generator to model the every single domain.

Concerning the age and gender combinations, we choose to model each gender combination of 3 different age intervals that are 0-17, 18-31 and 31-90. These intervals are selected taking two facts into consideration. First, we want to model discretized phases of human face during its aging from childhood to elderness. Second, we need to take imbalanced age distribution of UTKFace dataset into account, hence, select these age groups to prevent modeling an age interval better or worse than another. Consequently, we construct the architectures such that style encoder network extracting 6 different style vectors of the afore-mentioned age and gender combinations from an input image, mapping network that generates 6 style vectors from a random latent vector, and the discriminator as a multi-task network that outputs the real or fake score for each of the generated

combinations.

Since the generative model introduced in [59] is trained on CelebA-HQ dataset with over 30.000 images of size 256×256, we reduce the model capacity due to the fewer number of samples and the smaller image size in UTKFace dataset. Therefore, we employ three encoder and decoder blocks and two bottleneck blocks in the generator architecture, containing about half the size of the original network configuration. Besides, dimension of style vectors and hidden dimension of the mapping network are reduced to 16 and 256 respectively. All these design choices are made to prevent overfitting to the training data, as we require model to generalize well to infer on a completely new dataset it has never seen. As for the training configuration, we used Adam optimizer for parameter updates, learning rate of 10^{-6} for the mapping network and 10^{-4} for the discriminator, encoder and generator networks. Weight decay of 10^{-4} is used to help better generalization and the batch size is selected as 8 for the training. Once we train the model, we freeze the weights and obtain the generative network denoted as G in Figure 2.1.

In order to implement the kinship models K_n, we train them separately using a particular age and gender combination of the input pair that is generated by G. This help each kin model to specialize on a specific age and gender combination, modeling the age-and-gender-specific kinship features of that specific combination. Training configuration is held the same as described in Section 3.2 for each K_n.

Before aggregating all these age and gender combinations for kinship verification, we first analyze the individual impact of each of these age and gender normalized pairs on verification. First two columns of the Table 3.5 denote the transformed gender and age attributes of the normalized pairs respectively. Experiment results in the table show that normalization at any age and gender combination increase the overall kinship verification performance on both datasets compared to the baseline. Consistent with the previous experiments, improvements are more obvious for the KFW-II dataset compared to KFW-I. In terms of kinship verification at KFW-I, we observe that the normalization to age interval 31-90 outperforms the rest of the age groups with any gender combination. By

contrast, this is not the case for KFW-II as normalization to the younger age intervals yielded better verification performance. This might point out to the different age distribution of the subjects in these datasets rather than the modeling performance of the generator. Therefore, we might infer that the age distribution of subjects in KFW-I is relatively older compared to the KFW-II, considering the best verification accuracy obtained by normalization at older ages on KFW-I and younger ages on KFW-II. In order to validate that interpretation, we employ a pre-trained age regression network [72] to compute the age difference between the datasets. Supporting our reasoning, the inference results revealed an average of 9.5 years age difference between the KFW-I and KFW-II.

Focusing on the improvements in pairs with dissimilar genders, normalization to 18-30 years old male results in best performance in F-D verification on both datasets compared to the other combinations. For M-S verification on the other hand, normalization to 31-90 years old male considerably surpasses the remaining combinations at KFW-I, and yields in almost the same accuracy with the best performing normalization at 0-17 years male with only 0.2% of an accuracy difference. Since the improvements in the verification of kin pairs with dissimilar genders is notably higher when we normalized subjects to the male domain, the reason can be explained as the generator learned the male domain better than the female.

		KFW-I					KFW-II				
Gender	Age	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
M	0-17	0.6724	0.7437	0.6924	0.6025	0.6778	0.7180	0.7640	0.7740	0.7900	0.7615
M	18-30	0.6907	0.7083	0.7007	0.6324	0.6830	0.7420	0.7700	0.7740	0.7800	0.7665
M	31-90	0.6682	0.7148	0.7244	0.6543	0.6905	0.7380	0.7600	0.7560	0.7880	0.7605
F	0-17	0.6682	0.7310	0.7239	0.6018	0.6812	0.7320	0.7580	0.7820	0.7820	0.7635
F	18-30	0.6499	0.7120	0.7399	0.6413	0.6858	0.7320	0.7560	0.7720	0.7660	0.7565
F	31-90	0.6724	0.7183	0.7716	0.6196	0.6955	0.7320	0.7540	0.7900	0.7760	0.7630

Table 3.5: Performance of individual kin models trained on a specific age and gender combination. M denotes male and F denotes female.

Concerning the pairs with similar genders at KFW-I, we observe that all the age and gender combinations decrease the verification accuracy of F-S pair compared to the baseline, except the normalization at age interval of 0-17 years old. Given that, we can assume the age distribution of F-S pairs might be closer to the 0-17 years old age interval so that normalization within this interval help increase the verification performance. A similar pattern is observed for the M-D pair in which we obtain higher verification accuracies as the normalization at older age intervals further improve the performance, hence the reason can be explained as the age distribution of M-D pairs is older than the F-S pairs. For similar genders at KFW-II, we also observe that normalization at younger ages results in higher accuracy for F-S verification, whereas normalization at older age interval yields a better performance in M-D verification so that we can make a similar inference. Using the same pre-trained age regression network [72] we employed in validating the average age difference between the two datasets, we also confirm that the parents in F-S pairs are 6.4 and 10.2 years younger than the parents in M-D pairs in average, concerning the KFW-I and KFW-II datasets respectively.

Up to this point, we analyzed the improvement in kinship verification employing different age and gender combinations individually. Since we observe a solid increase in verification performance with all the normalization values, we then aggregate all of these age and gender versions. That way, we expect to benefit from every single age and gender combination of the input image pair simultaneously when predicting the kinship probability. To analyze the combined effect of all the age and gender normalized versions in kinship verification, we first experiment with rather naive approaches such as aggregating the kinship probabilities each K outputs, by means of majority voting, joint posterior, and the average posterior. For predicting the kinship score using the majority voting, all the kin models K predict a kinship probability using the corresponding age and gender combination on which they are trained for, then the input pair is inferred as kin if the majority of the models output a kinship probability that is greater than 0.5. Concerning the joint and average posteriors on the other hand, since the output of each kin model is an independent posterior probability, we infer the final kinship probability as either the product of these posteriors or the mean of

them respectively.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Mean Kinship Score	0.6644	0.7310	0.7439	0.6283	0.6919	0.7200	0.7640	0.7700	0.7680	0.7555
Majority Voting	0.6940	0.7115	0.7323	0.6509	0.6972	0.7420	0.7740	0.7760	0.8000	0.7729
Joint Posterior	0.6978	0.7179	0.7480	0.6509	0.7036	0.7400	0.7760	0.7840	0.7980	0.7745
Average Posterior	0.7015	0.7211	0.7441	0.6552	0.7055	0.7420	0.7780	0.7840	0.7940	0.7745

Table 3.6: Different approaches for kin score aggregation of each kin model

Experiment results of these approaches are reported in the table above. As expected, we observe that using all the age and gender normalization combinations led to an improvement of 1% to 3% in kinship verification accuracy compared to the verification using individual combination of age and gender normalization that are shown in Table 3.5. Moreover, considering the performance of different aggregation methods, majority voting performs worse than the joint and average posteriors on both datasets even if by a quite small margin. Therefore we can conclude from the results in Table 3.6 that utilizing all the age and gender combinations may result in a more robust kinship verification, while there are not any significant difference between the aggregation methods that we experiment.

As we observe certain improvements in verification after combining all the age and gender versions, we finally experiment with the pair-specific weighting of the kin models that are trained on particular age and gender normalized versions of the input pair as proposed in Section 2.2. To this end, instead of combining the age and gender versions by the naive approaches as discussed above, we implement the attention networks A in Figure 2.1 to obtain a pair-specific and dynamic combination of each kinship score that the kin models output.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Baseline	0.6608	0.7309	0.7207	0.5897	0.6755	0.6800	0.7140	0.6860	0.7060	0.6965
Attention on Original Data	0.6685	0.7309	0.7324	0.6472	0.6947	0.6950	0.7200	0.7040	0.7240	0.7107
Attention on Normalized Data	0.7090	0.7214	0.7436	0.6720	0.7115	0.7580	0.7880	0.7920	0.8020	0.7850

Table 3.7: Kinship verification performance of pair-specific attention on age and gender combinations

Experiment results for pair-specific weighting of kin models are shown in the table above. Besides a minor improvement in the overall verification accuracy compared to the results shown in Table 3.6, employing pair-specific attention further reduce the verification performance gap between the similar and dissimilar genders on both datasets. The reason can be explained as the attention network learns which kin model should have the most influence on output according to the different age and gender features extracted by each of the kin models. Therefore it is expected to improve the verification performance of the pairs with relatively lower accuracy, since the network emphasize on the most eligible age and gender normalized version of the input pair by assigning a higher weight than the rest of the age and gender combinations. While adding only the attention mechanism to the training of the network using the original data results in a slight improvement compared to the baseline, that enhancement in kinship verification performance considerably increases when the age and gender combinations are employed in the training, approving the effect of age and gender in kinship verification.

3.5.2 Normalization in Latent Space

In addition to our study on age and gender normalization in image space, we analyze the effect of normalization in latent space to kinship verification. To this end, we use the same generative modeling backbones that we analyzed in Section 3.5.1 to obtain comparable results.

Recall that for achieving the sample-specific normalization in image space, we minimize the loss function in Equation 2.16, where we compute the age and gender losses using the inference of networks F_{a} and F_{g} in image space. To realize sample-specific normalization in the latent space instead, we follow the implementation shown in Figure 2.10 and described in Section 2.1.3.

For convenience, we do not experiment with all the different age groups that are studied in Table 3.4 and keep only the best performing normalization scheme where the subjects are normalized to the neutral gender and to the age of 35. Furthermore, we perform some further analysis to investigate the possible deficiencies in identity information that might have been introduced during the age and gender normalization. To this end, we conduct three additional experiments as the following. First, we amplify the influence of perceptual loss in the overall normalization loss in Equation 2.16 by increasing its coefficient. Second, we employ an additional loss term that is the cosine similarity between the original image and its normalized version computed in the latent space. Third, we increase the coefficients of both perceptual and cosine similarity losses.

Results of the aforementioned experiments are reported in Table 3.8. As we can see, normalization in latent space provides almost the same overall improvement in kinship verification accuracy compared to the same normalization in image space as shown in the third row of Table 3.4. On the other hand, experiments with different loss function configurations show that there is a trade-off between the normalization and preserving the facial identity information more aggressively using the loss coefficients. The most visible effect of increasing the coefficient of the perceptual loss is that it yields a performance drop in M-S verification at KFW-I and an increase in M-D verification at KFW-II. This might be indicating that the normalization of age and gender does not degrade the identity information in general, and solely increasing the influence of perceptual loss only suppresses the efficiency of the normalization.

Employing the additional loss term cosine similarity results in a considerable performance decrease in F-D verification while an increase in M-S verification. Recall that the perceptual loss is computed using the feature maps of a neural

network while the cosine similarity loss is obtained directly in the latent space. Since cosine similarity in latent space mostly indicates the encoded facial similarity in the embedded space of the generator, we can explain the contradicting effect in F-D and M-S pairs as the side effect of normalization in corrupting the identity information for the M-S pair. As for the F-D pair, normalization yields rather the expected improvement than the undesired effect of repressing the identity-related information.

	KFW-I					KFW-II				
	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Gender and Age Normalized	0.7165	0.7275	0.7487	0.6853	0.7195	0.7840	0.7940	0.7960	0.7960	0.7925
+ Higher Perceptual Loss	0.7205	0.7213	0.7481	0.6681	0.7145	0.7820	0.7900	0.8060	0.7920	0.7925
+ Cosine Similarity	0.6868	0.7212	0.7210	0.7109	0.7100	0.7780	0.7980	0.8060	0.7960	0.7945
+ Higher Cosine Similarity and Perceptual Loss	0.7316	0.7216	0.7561	0.6721	0.7204	0.7900	0.7980	0.8100	0.7840	0.7955

Table 3.8: Sample-specific normalization of age and gender in latent space

Combined effect of employing the cosine similarity and increasing the perceptual loss coefficient provides slightly better verification performance, especially for the pairs F-D and M-D.

Lastly, we analyze the effect of normalization in latent space using the StarGAN backbone for generative modeling. To this end, we interpolate between the genders and an age style vector in latent space to find a brand new style vector which encodes the style representation of the input subject that is age and gender normalized. This make sense since each style vector contains semantic information about the specific characteristics of the face, therefore, we can compute the neutral gender version of each age group unlike the experiments in Table 3.5 where we need to combine a specific gender with an age group.

Experiment results in Table 3.9 shows that normalization in latent space also enhances the kinship verification performance of the baseline on both datasets. Although we cannot directly compare the results with normalization in image space due to the normalized gender, we can still draw some insights by analyzing

these experiments.

Concerning the experiments on KFW-I, normalization in latent space in general leads to better verification accuracies compared to the different age and gender normalization combinations shown in Table 3.5. Verification of the pairs with dissimilar gender is higher than the most combinations in Table 3.5, as we might expect since the neutral gender representation in latent space is utilized. For the M-D pair, normalization in latent space results in always a better verification except the normalization to 31-90 years old female in image space. Nevertheless, we see the same pattern in normalization at latent space as the M-D pair tends to give better results when normalized to the older ages. Similarly, F-S pair verification is consistently better when the subjects are normalized to younger ages in both image space and latent space normalization.

		KFW-I					KFW-II				
Gender	Age	F-D	F-S	M-D	M-S	Mean	F-D	F-S	M-D	M-S	Mean
Neutral	0-17	0.6721	0.7410	0.7441	0.6413	0.6996	0.7680	0.7540	0.7620	0.7600	0.7610
Neutral	18-30	0.6687	0.7379	0.7407	0.6510	0.6995	0.7600	0.7420	0.7780	0.7480	0.7580
Neutral	31-90	0.6778	0.7380	0.7527	0.6673	0.7089	0.7580	0.7400	0.7740	0.7550	0.7570

Table 3.9: Age and gender normalization by style vector interpolation in StarGAN latent space

Considering the experiments on KFW-II, normalization to neutral gender in latent space increases the verification performance for F-D pair, but decreases it for the M-S pair for all the age groups. Although the M-S pair normalization degrades the verification accuracy compared to the image space normalization, the results are consistent by means of improvement in M-S accuracy when normalized to the 0-17 age group. Concerning the F-S and M-D verification, normalization to neutral gender in latent space does not show any significant improvement as expected, since they are already on the same surface in terms of gender. Again,

results show consistent improvement changes in terms of the age, as the normalization to younger age for F-S and to older age for M-D pairs leads to a relatively better verification.

	KFW-I	KFW-II
Baseline	0.6755	0.6965
Latent Projection	0.6714	0.6962
Style Modification	0.6589	0.6905
Sample-Specific Optimization	**0.7178**	**0.7970**
Pair-Specific Weighting	**0.7115**	**0.7850**

Table 3.10: Summary of change in kinship verification accuracy employing the proposed normalization methods

All in all, we observe that the methods which employ vector operations in the latent space results in a very limited change or even reduced the accuracy due to altering the identity-related facial characteristics. On the other hand, age and gender normalization by sample-specific optimization and pair-specific weighting yields a considerable increase in kinship verification performance compared to the baseline.

Chapter 4

Conclusion

The individual and combined impact of age- and gender-related facial attributes in kinship verification have been investigated. Several methods have been proposed to eliminate their undesired effect in this context. The proposed approach utilizes the extensive modeling capabilities of generative adversarial networks to model and removes the age and gender attributes, enhancing the verification performance of an arbitrary kinship model on two benchmark datasets without making any changes in the hyperparameters or the architecture.

We have first proposed normalizing the gender-specific features from the facial images by learning a transformation between the male and female gender domains, then finding an equally-spaced latent dimension to represent the faces in a gender-neutralized manner. To do so, we have used the neutral gender representation in latent space and projected the subjects' faces onto this representation to obtain their gender-neutralized versions.

In another approach, inspired by the style transfer literature, we have extracted the style vectors of faces and synthesized gender-neutral versions of the same identities by combining these style vectors with the ones extracted from the neutral-gender representation. This is an intuitive way of mixing diverse facial attributes that blend facial identity styles with neutral gender styles.

Moreover, we defined generating an age-and-gender-normalized version of the subject as an optimization problem. To this end, we iteratively generate a normalized version of the input face, minimizing the proposed age and gender normalization loss. The overall normalization loss involves perceptual similarity and the age and gender normalization loss terms to preserve the identity information at a reasonable scale while yielding the desired normalization of these two attributes. The age and gender normalization losses are computed by employing auxiliary networks' logits that perform age regression and gender classification, respectively. These inferences are computed both in the image and latent spaces in order to compare their efficiency against each other. Image-space models are trained on public datasets that contain age and gender labels, while the models that perform inference in latent space are trained in the latent space of the corresponding pre-trained generator. Normalizing the age and gender with this method has achieved the best kinship verification performance due to the sample-specific optimization of the images. However, this is expectedly the most computationally costly method among all the proposed approaches.

Furthermore, the normalization performance is subject to the performance of age and gender networks that are employed to compute the corresponding loss terms. Our experiments with normalizing only the age or gender using this approach showed that the individual contribution of age or gender normalization contributes to the performance improvement in kinship verification in a similar magnitude. Concerning the benefit of computing the normalization loss in the image or the latent space, we observed that the normalization in the latent space slightly provided better performance in terms of kinship verification.

Unlike the previous approaches in which we directly normalized the age and gender attributes, we performed kinship verification utilizing the subjects' pair-specific weighted age and gender combinations. Therefore, we have modeled both genders along with the three different age groups using a multi-domain generator. Synthetic age and gender combinations of the image pairs are used for pre-training specialized kinship models, which are then trained using an attention network that assigns weights to these combinations according to the extracted kinship features from each pair. That way, the model dynamically determines how much each age

and gender combination of the input pair would contribute to the final kinship score. Extensive experiments are conducted to validate the efficiency of this approach by analyzing the individual kinship models that are trained on different age and gender combinations. Each model increased the verification accuracy at various scales, improving even more when the attention model is attached to combine all the information out of different age and gender combinations of the input face pair. This approach yielded the second-best improvement in kinship verification after the sample-specific normalization of age and gender attributes.

All in all, we have shown that the differences stemmed from age- and gender-related facial attributes degrade the kinship verification performance. By normalizing the gender, we have reduced the verification performance gap between the similar and distinct gender kin pairs by about 6% on KFW-I. Furthermore, combined normalization of age and gender has improved overall kinship verification accuracy up to 10% on KFW-II. To the best of our knowledge, this is the first study that comprehensively explores the impact of age and gender in kinship verification and proposes several methods to remove their degrading effect on kinship verification performance. Future research approaches might include finer-grained modeling of age and gender to improve the separation of these characteristics from kinship-related features.

References

[1] Rebecca L Burch and Gordon G Gallup. *Perceptions of paternal resemblance predict family violence*. Tech. rep.

[2] Paola Bressan and Maria F Dal Martello. *TALIS PATER, TALIS FILIUS: Perceived Resemblance and the Belief in Genetic Relatedness*. Tech. rep. 3. 2002.

[3] Steven M. Platek et al. "Where am I? The neurological correlates of self and other". In: *Cognitive Brain Research* 19.2 (2004), pp. 114–122. ISSN: 09266410. DOI: 10.1016/j.cogbrainres.2003.11.014.

[4] Maria F. Dal Martello and Laurence T. Maloney. "Lateralization of kin recognition signals in the human face". In: *Journal of Vision* 10.8 (2010), pp. 1356–1366. ISSN: 15347362. DOI: 10.1167/10.8.9.

[5] Haibin Yan and Chaohui Song. "Multi-scale deep relational reasoning for facial kinship verification". In: *Pattern Recognition* 110 (2021), p. 107541. ISSN: 00313203. DOI: 10.1016/j.patcog.2020.107541. URL: https://doi.org/10.1016/j.patcog.2020.107541.

[6] Naman Kohli, Richa Singh, and Mayank Vatsa. "Self-similarity representation of Weber faces for kinship classification". In: *2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2012* (2012), pp. 245–250. DOI: 10.1109/BTAS.2012.6374584.

[7] Gowri Somanath and Chandra Kambhamettu. "Can faces verify blood-relations?" In: *2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2012* (2012), pp. 105–112. DOI: 10.1109/BTAS.2012.6374564.

[8] Hamdi Dibeklioglu. "Visual Transformation Aided Contrastive Learning for Video-Based Kinship Verification". In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob (2017), pp. 2478–2487. ISSN: 15505499. DOI: 10.1109/ICCV.2017.269.

[9] Naman Kohli et al. "Hierarchical representation learning for kinship verification". In: *IEEE Transactions on Image Processing* 26.1 (2017), pp. 289–302. ISSN: 10577149. DOI: 10.1109/TIP.2016.2609811. arXiv: 1805.10557.

[10] Siyu Xia, Ming Shao, and Yun Fu. "Kinship verification through transfer learning". In: *IJCAI International Joint Conference on Artificial Intelligence* (2011), pp. 2539–2544. ISSN: 10450823. DOI: 10.5591/978-1-57735-516-8/IJCAI11-422.

[11] Haibin Yan, Xiuzhuang Zhou, and Yongxin Ge. "Neighborhood repulsed correlation metric learning for kinship verification". In: *2015 Visual Communications and Image Processing, VCIP 2015* (2016), pp. 2594–2601. DOI: 10.1109/VCIP.2015.7457930.

[12] Haibin Yan, Xiuzhuang Zhou, and Yongxin Ge. "Neighborhood repulsed correlation metric learning for kinship verification". In: *2015 Visual Communications and Image Processing, VCIP 2015* 36.2 (2016), pp. 331–345. DOI: 10.1109/VCIP.2015.7457930.

[13] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. "Tri-Subject Kinship Verification: Understanding the Core of A Family". In: *IEEE Transactions on Multimedia* 17.10 (2015), pp. 1855–1867. ISSN: 15209210. DOI: 10.1109/TMM.2015.2461462. arXiv: 1501.02555.

[14] Ruogu Fang et al. "Kinship classification by modeling facial feature heredity". In: *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings* (2013), pp. 2983–2987. DOI: 10.1109/ICIP.2013.6738614.

[15] Ruogu Fang et al. "Towards computational models of kinship verification". In: *Proceedings - International Conference on Image Processing, ICIP* (2010), pp. 1577–1580. ISSN: 15224880. DOI: 10.1109/ICIP.2010.5652590.

[16] Ming Shao, Siyu Xia, and Yun Fu. "Genealogical face recognition based on UB KinFace database". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2011), pp. 60–65. ISSN: 21607516. DOI: 10.1109/CVPRW.2011.5981801.

[17] Hamdi Dibeklio. "Are You Really Smiling at Me ? Spontaneous versus Posed Enjoyment Smiles". In: *Eccv* (2012).

[18] Haibin Yan et al. "Discriminative multimetric learning for kinship verification". In: *IEEE Transactions on Information Forensics and Security* 9.7 (2014), pp. 1169–1178. ISSN: 15566013. DOI: 10.1109/TIFS.2014.2327757.

[19] Min Xu and Yuanyuan Shang. "Kinship Measurement on Face Images by Structured Similarity Fusion". In: *IEEE Access* 4 (2016), pp. 10280–10287. ISSN: 21693536. DOI: 10.1109/ACCESS.2016.2635147.

[20] Shahar Mahpod and Yosi Keller. "Kinship verification using multiview hybrid distance learning". In: *Computer Vision and Image Understanding* 167.September 2017 (2018), pp. 28–36. ISSN: 1090235X. DOI: 10.1016/j.cviu.2017.12.003. URL: https://doi.org/10.1016/j.cviu.2017.12.003.

[21] Xiuzhuang Zhou et al. "Kinship verification from facial images under uncontrolled conditions". In: *MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops* (2011), pp. 953–956. DOI: 10.1145/2072298.2071911.

[22] G. Guo and Xiaolong Wang. "Kinship Measurement on Salient Facial Features". In: *IEEE Transactions on Instrumentation and Measurement* 61 (2012), pp. 2322–2325.

[23] Xiuzhuang Zhou et al. "Gabor-based gradient orientation pyramid for kinship verification under uncontrolled environments". In: *MM 2012 - Proceedings of the 20th ACM International Conference on Multimedia* (2012), pp. 725–728. DOI: 10.1145/2393347.2396297.

[24] Siyu Xia, Ming Shao, and Yun Fu. "Toward kinship verification using visual attributes". In: *Proceedings - International Conference on Pattern Recognition* Icpr (2012), pp. 549–552. ISSN: 10514651.

[25] Ajit Puthenputhussery, Qingfeng Liu, and Chengjun Liu. "SIFT flow based genetic fisher vector feature for kinship verification". In: (2016), pp. 2921–2925. DOI: 10.1109/ICIP.2016.7532894.

[26] Zhifei Zhang, Yang Song, and Hairong Qi. "Age progression/regression by conditional adversarial autoencoder". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 4352–4360. DOI: 10.1109/CVPR.2017.463. arXiv: 1702.08423.

[27] Afshin Dehghan et al. "Who do i look like? Determining parent-offspring resemblance via gated autoencoders". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014), pp. 1757–1764. ISSN: 10636919. DOI: 10.1109/CVPR.2014.227.

[28] Mengyin Wang et al. "Deep kinship verification". In: *2015 IEEE 17th International Workshop on Multimedia Signal Processing, MMSP 2015* (2015). DOI: 10.1109/MMSP.2015.7340820.

[29] Lei Li et al. "Kinship Verification from Faces via Similarity Metric Based Convolutional Neural Network". In: (). DOI: 10.1007/978-3-319-41501-7.

[30] Jiwen Lu et al. "Neighborhood repulsed metric learning for kinship verification". In: *IEEE transactions on pattern analysis and machine intelligence* 36.2 (2013), pp. 331–345.

[31] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks". In: *33rd International Conference on Machine Learning, ICML 2016* 4 (2016), pp. 2611–2620. arXiv: 1601.06759.

[32] Aäron Van Den Oord et al. "Conditional image generation with PixelCNN decoders". In: *Advances in Neural Information Processing Systems* (2016), pp. 4797–4805. ISSN: 10495258. arXiv: 1606.05328.

[33] Diederik P. Kingma and Max Welling. "Auto-encoding variational bayes". In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* Ml (2014), pp. 1–14. arXiv: 1312.6114.

[34] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2014), pp. 139–144. ISSN: 15577317. DOI: 10.1145/3422622. arXiv: 1406.2661.

[35] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* (2016), pp. 1–16. arXiv: 1511.06434.

[36] Yujun Shen et al. "Interpreting the latent space of GANs for semantic face editing". In: *arXiv* (2019), pp. 9243–9252. ISSN: 23318422.

[37] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. "DNA-GAN: Learning disentangled representations from multi-attribute images". In: *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings* (2018), pp. 1–14. arXiv: 1711.05415.

[38] Jiapeng Zhu et al. "Disentangled Inference for GANs with Latently Invertible Autoencoder". In: (2019). arXiv: 1906.08090. URL: http://arxiv.org/abs/1906.08090.

[39] Nicki S. Detlefsen and Søren Hauberg. "Explicit disentanglement of appearance and perspective in generative models". In: *Advances in Neural Information Processing Systems* 32.NeurIPS (2019). ISSN: 10495258. arXiv: 1906.11881.

[40] Kanglin Liu et al. "Disentangling Latent Space for Unsupervised Semantic Face Editing". In: (2020), pp. 1–11. arXiv: 2011.02638. URL: http://arxiv.org/abs/2011.02638.

[41] Erik Härkönen et al. "GANSpace: Discovering Interpretable GAN Controls". In: NeurIPS (2020), pp. 1–10. ISSN: 10495258. arXiv: 2004.02546. URL: http://arxiv.org/abs/2004.02546.

[42] Xinqi Zhu, Chang Xu, and Dacheng Tao. "Learning Disentangled Representations with Latent Variation Predictability". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12355 LNCS (2020), pp. 684–700. ISSN: 16113349. DOI: 10.1007/978-3-030-58607-2_40. arXiv: 2007.12885.

[43] Yu Deng et al. “Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 5153–5162. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00520. arXiv: 2004.11660.

[44] Andrey Voynov and Artem Babenko. “Unsupervised discovery of interpretable directions in the gan latent space”. In: *37th International Conference on Machine Learning, ICML 2020* PartF168147-13 (2020), pp. 9728–9738. arXiv: 2002.03754.

[45] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “Image Style Transfer Using Convolutional Neural Networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (2016), pp. 2414–2423. ISSN: 10636919. DOI: 10.1109/CVPR.2016.265.

[46] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization”. In: 2016 (2016). arXiv: 1607.08022. URL: http://arxiv.org/abs/1607.08022.

[47] Xun Huang and Serge Belongie. “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob (2017), pp. 1510–1519. ISSN: 15505499. DOI: 10.1109/ICCV.2017.167. arXiv: 1703.06868.

[48] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), pp. 1–14. arXiv: 1409.1556.

[49] Tero Karras, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (2019), pp. 4396–4405. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00453. arXiv: 1812.04948.

[50] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (2017), pp. 5967–5976. DOI: 10.1109/CVPR.2017.632. arXiv: 1611.07004.

[51] Jun Yan Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob (2017), pp. 2242–2251. ISSN: 15505499. DOI: 10.1109/ICCV.2017.244. arXiv: 1703.10593.

[52] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9906 LNCS (2016), pp. 694–711. ISSN: 16113349. DOI: 10.1007/978-3-319-46475-6_43. arXiv: 1603.08155.

[53] Chuan Li and Michael Wand. “Precomputed real-time texture synthesis with markovian generative adversarial networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9907 LNCS (2016), pp. 702–716. ISSN: 16113349. DOI: 10.1007/978-3-319-46487-9_43. arXiv: 1604.04382.

[54] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (2017), pp. 105–114. DOI: 10.1109/CVPR.2017.19. arXiv: 1609.04802.

[55] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (2016), pp. 770–778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[56] Tero Karras et al. “Progressive growing of gans for improved quality, stability, and variation”. In: *arXiv* (2017), pp. 1–26. ISSN: 23318422. arXiv: 1710.10196.

[57] Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: (2014), pp. 1–7. arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

[58] Tero Karras et al. “Analyzing and improving the image quality of stylegan”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 8107–8116. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00813. arXiv: 1912.04958.

[59] Yunjey Choi et al. “StarGAN v2: Diverse Image Synthesis for Multiple Domains”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 8185–8194. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00821. arXiv: 1912.01865.

[60] Yunjey Choi et al. “StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 8789–8797. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00916. arXiv: 1711.09020.

[61] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. “Which training methods for GANs do actually converge?” In: *International conference on machine learning*. PMLR. 2018, pp. 3481–3490.

[62] Ming-Yu Liu et al. *Few-Shot Unsupervised Image-to-Image Translation*. 2019. arXiv: 1905.01723 [cs.CV].

[63] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2018), pp. 586–595. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00068. arXiv: 1801.03924.

[64] Rasmus Rothe, Radu Timofte, and Luc Van Gool. “DEX: Deep EXpectation of apparent age from a single image”. In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2015.

[65] A. Midori Albert, Karl Ricanek, and Eric Patterson. “A review of the literature on the aging adult skull and face: Implications for forensic science research and applications”. In: *Forensic Science International* 172.1 (2007), pp. 1–9. ISSN: 03790738. DOI: 10.1016/j.forsciint.2007.03.015.

[66] Wei Wang, Shaodi You, and Theo Gevers. "Kinship Identification Through Joint Learning Using Kinship Verification Ensembles". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12367 LNCS (2020), pp. 613–628. ISSN: 16113349. DOI: 10.1007/978-3-030-58542-6_37. arXiv: 2004.06382.

[67] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), pp. 1–15. arXiv: 1412.6980.

[68] Vahid Kazemi and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014), pp. 1867–1874. ISSN: 10636919. DOI: 10.1109/CVPR.2014.241.

[69] Jie Hu. "Squeeze-and-Excitation_Networks_CVPR_2018_paper.pdf". In: *Cvpr* (2018), pp. 7132–7141. URL: http://openaccess.thecvf.com/content%7B%5C_%7Dcvpr%7B%5C_%7D2018/html/Hu%7B%5C_%7DSqueeze-and-Excitation%7B%5C_%7DNetworks%7B%5C_%7DCVPR%7B%5C_%7D2018%7B%5C_%7Dpaper.html.

[70] Qiong Cao et al. "VGGFace2: A dataset for recognising faces across pose and age". In: *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018* (2018), pp. 67–74. DOI: 10.1109/FG.2018.00020. arXiv: 1710.08092.

[71] Roy Or-El et al. "Lifespan Age Transformation Synthesis". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12351 LNCS (2020), pp. 739–755. ISSN: 16113349. DOI: 10.1007/978-3-030-58539-6_44. arXiv: 2003.09764.

[72] Rasmus Rothe, Radu Timofte, and Luc Van Gool. "Deep expectation of real and apparent age from a single image without facial landmarks". In: *International Journal of Computer Vision* 126.2-4 (2018), pp. 144–157.

Appendix A

Supplementary Figures

Figures that show different stages of age and gender normalization methods are added to the appendix for reference.

Figure A.1: Age and gender combinations of UTKFace samples generated by StarGAN. First row is the original face images from the dataset.

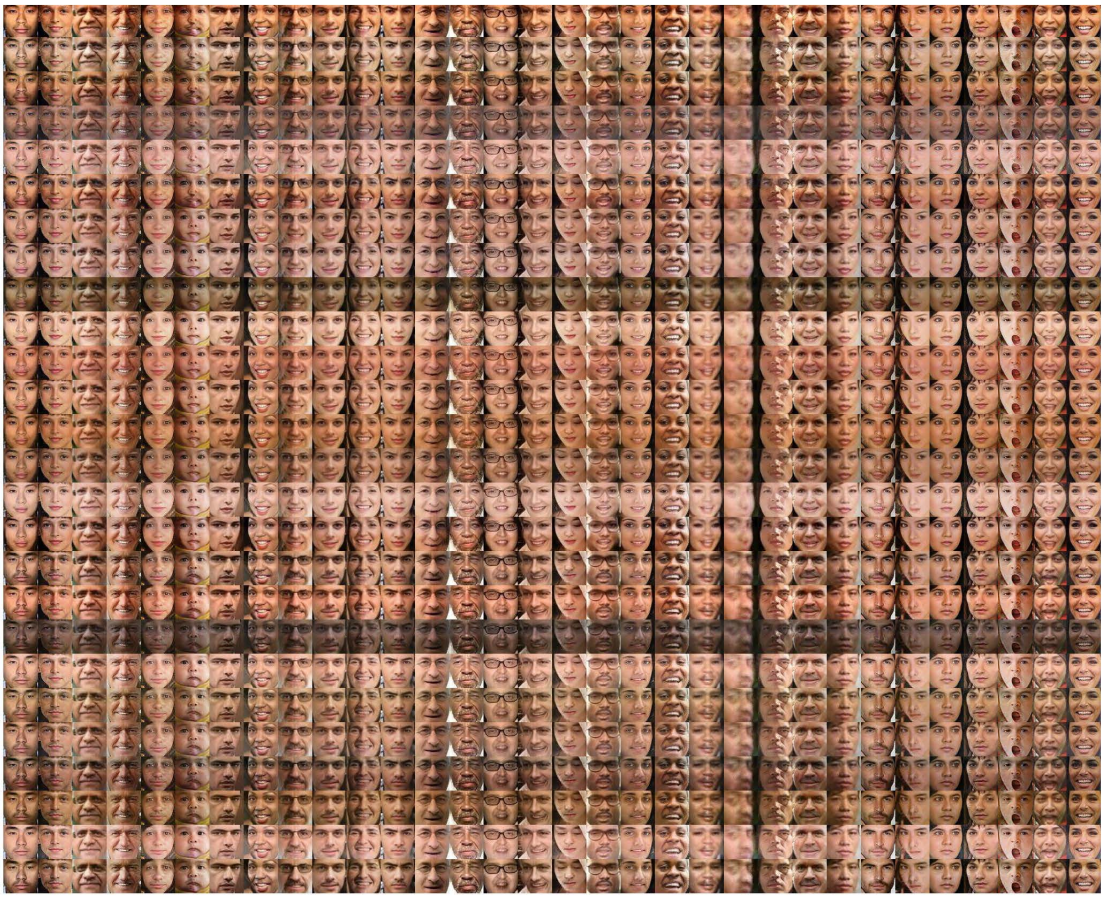


Figure A.2: Age and gender combinations of UTKFace samples generated by StarGAN continued.

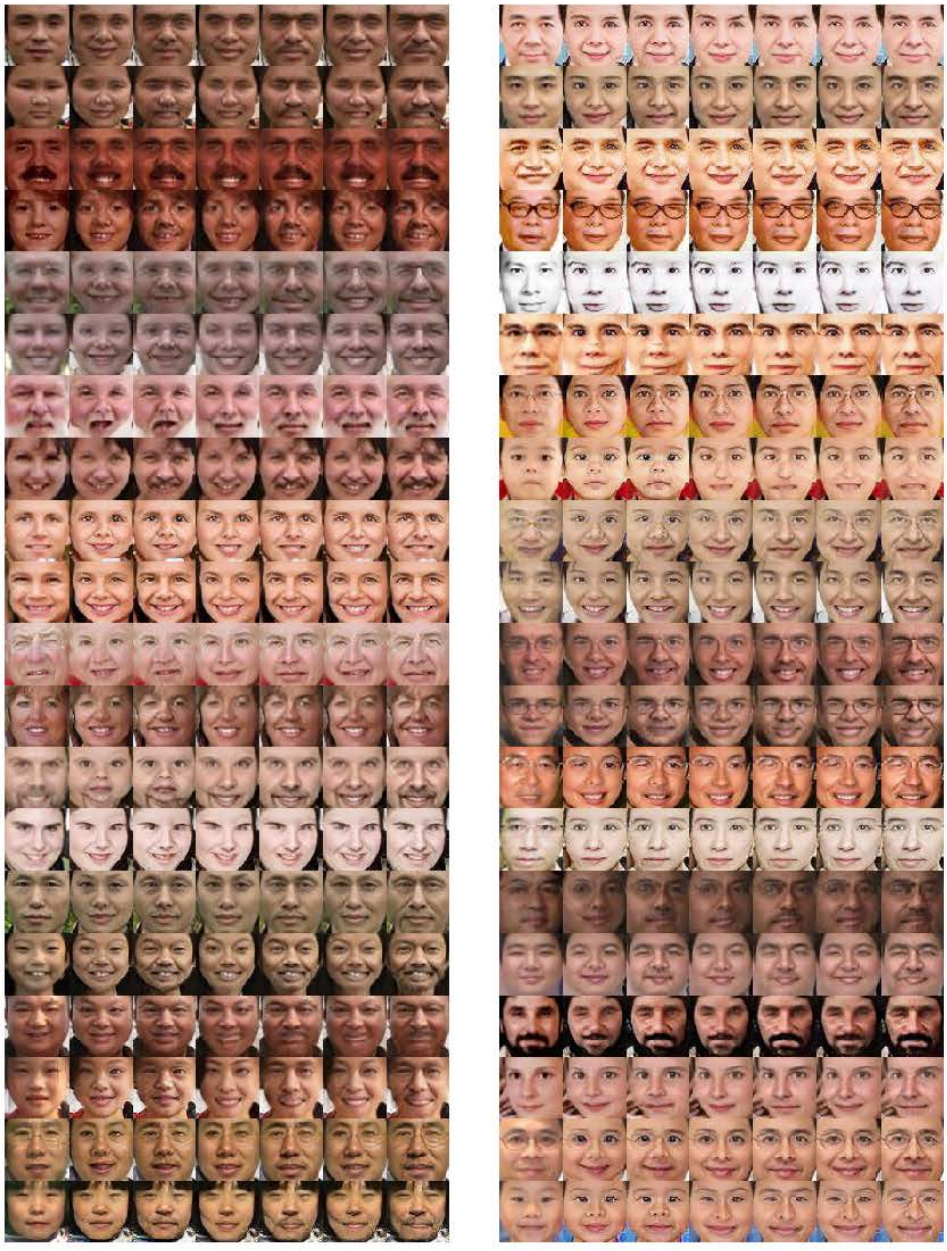


Figure A.3: Random F-D and F-S pair samples from KFW-I dataset that different age and gender combinations are generated. Each row contain one subject and their image in the following order. Original, 0-17 female, 0-17 male, 18-30 female, 18-30 male, 31-90 female, 31-90 male

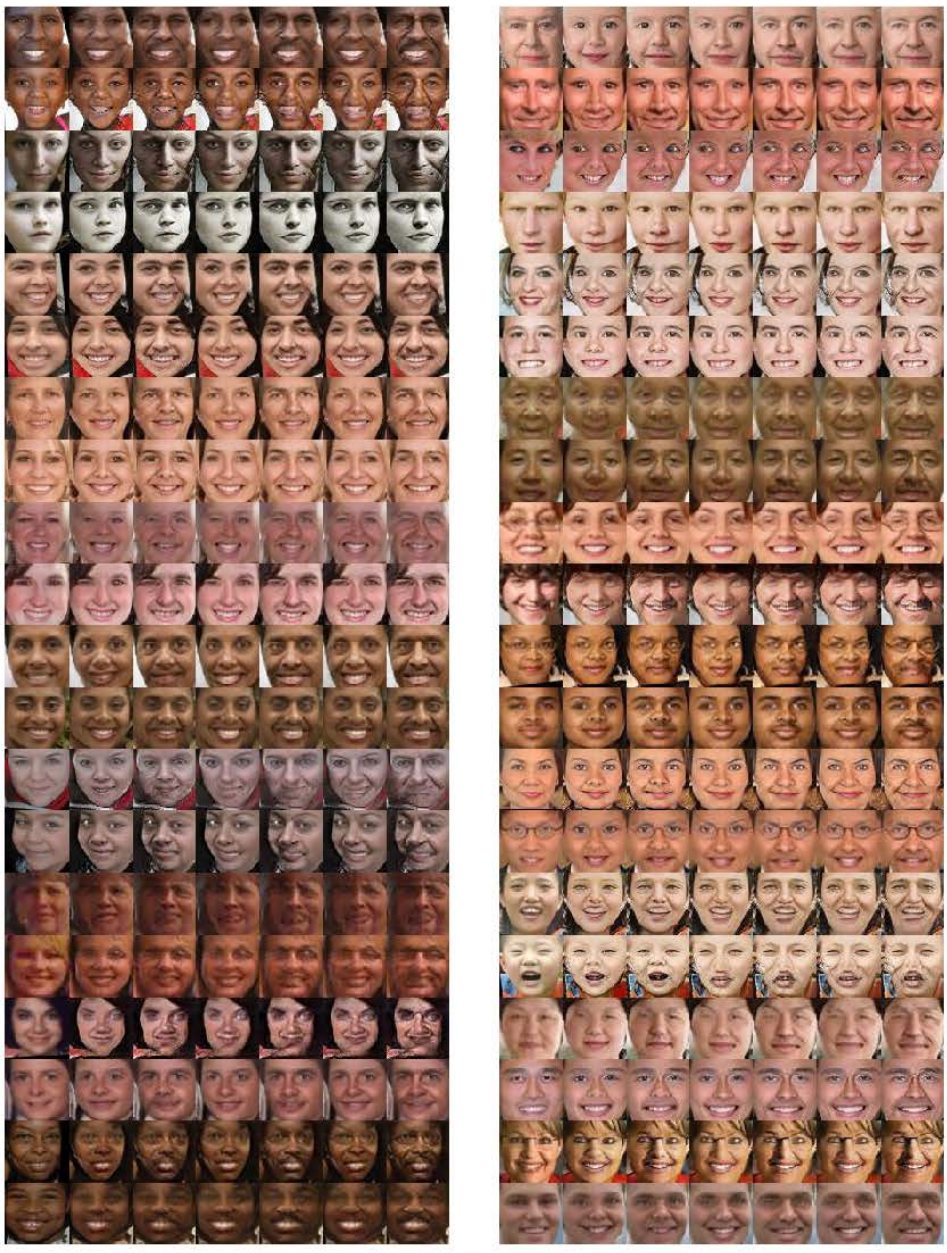


Figure A.4: Random M-D and M-S pair samples from KFW-I dataset that different age and gender combinations are generated. Each row contain one subject and their image in the following order. Original, 0-17 female, 0-17 male, 18-30 female, 18-30 male, 31-90 female, 31-90 male

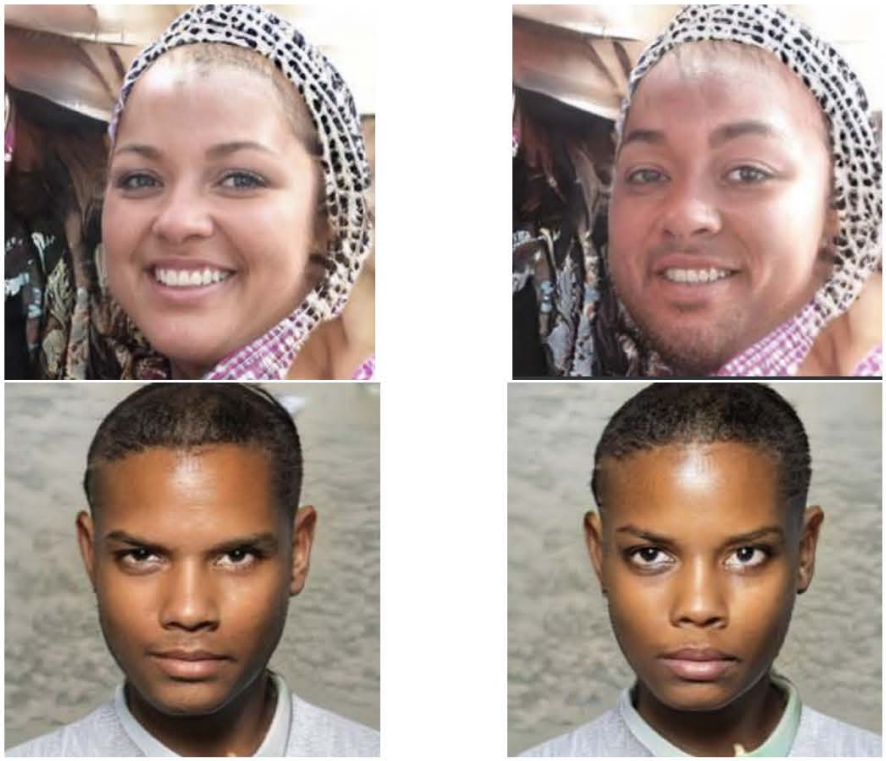


Figure A.5: Sample-specific gender normalization examples on randomly generated samples. Subjects are gender neutralized from the domains female and male as illustrated in the respective rows.

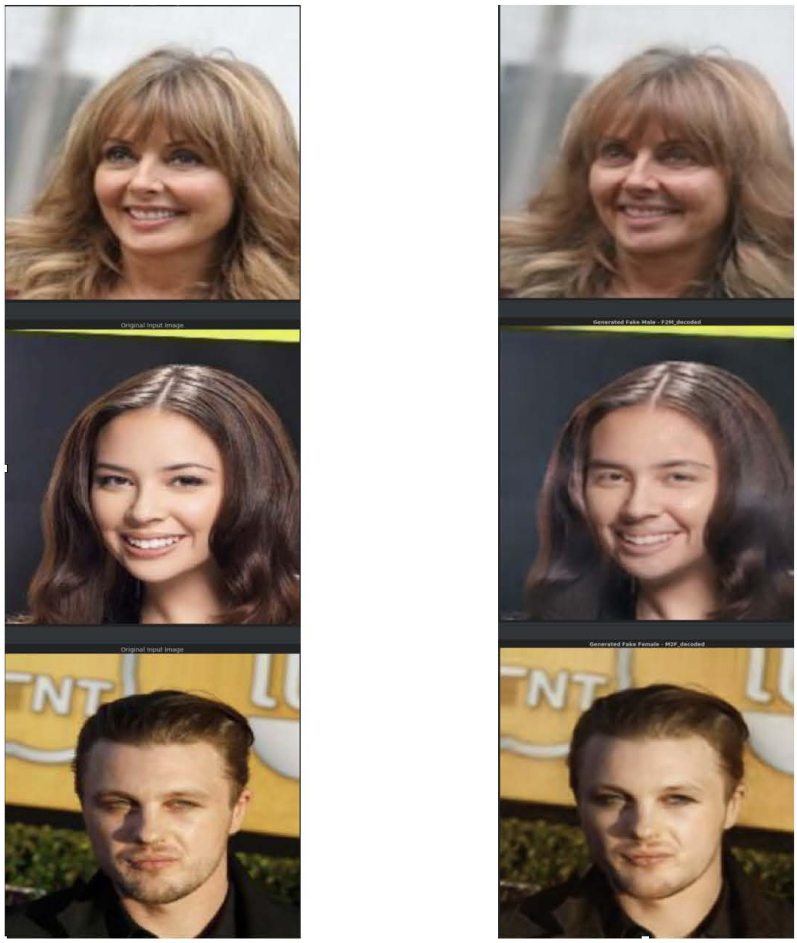


Figure A.6: Normalization by equally-spaced latent dimension finding using the CycleGAN backbone. Random samples from CelebA dataset are gender normalized to show the relatively small effect in verification. The effect of normalization is smaller than the other methods we proposed, hence it yielded a minimal effect in kinship verification as discussed in the experiments section.

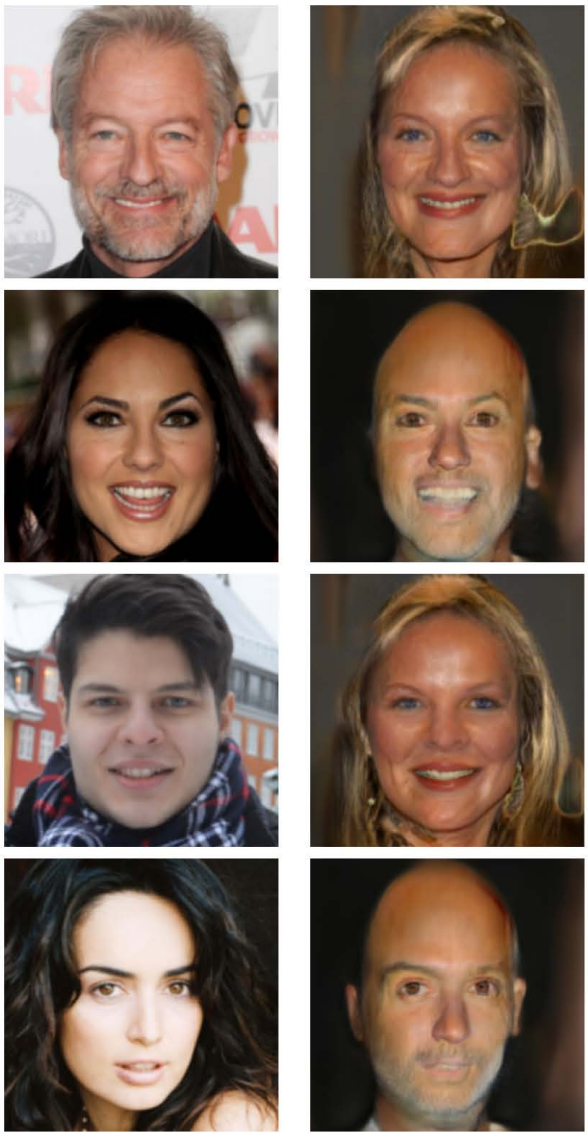


Figure A.7: Normalization by style vector modification. Random samples that are gender normalized to show the negative effect in verification. We observe that this normalization method results in a normalized faces in a relatively smaller space since all of the output are somehow similar in terms of the general facial shapes for both male and female domains. Therefore the effect in kinship verification has downgraded the performance as discussed in the experiments section.