Face Morphing: Fooling a Face Recognition System Is Simple!

Stefan Hörmann   Tianlin Kong   Torben Teepe   Fabian Herzog   Martin Knoche   Gerhard Rigoll
Technical University of Munich
s.hoermann@tum.de

Abstract

State-of-the-art face recognition (FR) approaches have shown remarkable results in predicting whether two faces belong to the same identity, yielding accuracies between 92% and 100% depending on the difficulty of the protocol. However, the accuracy drops substantially when exposed to morphed faces, specifically generated to look similar to two identities. To generate morphed faces, we integrate a simple pretrained FR model into a generative adversarial network (GAN) and modify several loss functions for face morphing. In contrast to previous works, our approach and analyses are not limited to pairs of frontal faces with the same ethnicity and gender. Our qualitative and quantitative results affirm that our approach achieves a seamless change between two faces even in unconstrained scenarios. Despite using features from a simpler FR model for face morphing, we demonstrate that even recent FR systems struggle to distinguish the morphed face from both identities obtaining an accuracy of only 55-70%. Besides, we provide further insights into how knowing the FR system makes it particularly vulnerable to face morphing attacks.

1. Introduction

Drawing from the impressive results of generative adversarial networks (GANs), face manipulation tasks have been investigated more frequently in the research community. Face manipulation is employed in multiple applications, *e.g.*, face swapping [24, 25, 30], face attribute manipulation [15], face beautification [5, 10], and (anti-)aging [15, 26, 33]. Face swapping targets substituting the identity of a face with the identity in a target image, maintaining background, head pose, and facial expression of the original image. Thus, identity features must be extracted and disentangled from the remaining information and introduced into the source image.

In contrast to face swapping, face morphing aims to create a seamless transition between two faces, X_1 and X_2, which involves identity, attributes, head pose, and background. Hence, when considering the information from
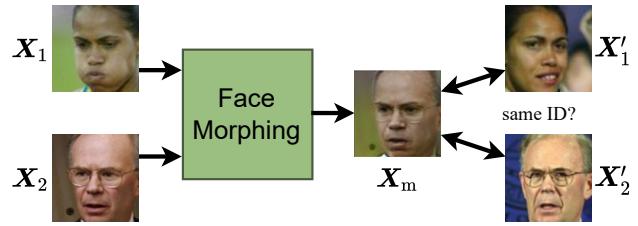

Figure 1. Example of a face morphing attack: The morphed face X_m is created given X_1 and X_2 from different identities. The objective is to determine whether an FR system classifies X_m to have the same identities as X_1' and X_2' despite X_1' and X_2' being from different identities.

both faces equally, the morphed face X_m looks similar to X_1 and X_2, as depicted in Fig. 1. To ensure that security-sensitive applications such as automatic border control or access control are not exposed to morphed faces, the employment of face recognition (FR) systems is typically accompanied by prior deepfake detection systems [34, 46] with the objective of detecting such tampered images. One popular approach in unlocking mobile phones is considering an additional infrared image, which makes it particularly challenging to create a suitable morphed face that matches the owner's infrared signature. However, if deepfake detection is not part of the FR system or fails to detect the morphed faces, it is crucial to determine how susceptible state-of-the-art FR systems are to such attacks.

Recent FR systems [8, 9, 22] report face verification results exceeding 99.5% on the arguably simple labeled faces in the wild (LFW) dataset [17], yet also reach 92% under more challenging cross-age and cross-pose scenarios. First analyses of FR performance under morphing attacks have been published before [23, 27, 35, 37, 39, 40, 44]. However, their results are limited as they only evaluate on frontal images with same gender and ethnicty [27, 35, 37, 39, 40, 44], only replace face parts [27, 35], or do not evaluate state-of-the-art FR methods [23, 27, 35, 37, 39]. To the best of our knowledge, no analysis has been published revealing the vulnerability of state-of-the-art FR systems on challenging datasets comprising images taken in the wild.

Our contributions can be summed up as follows:

- We show how a pretrained FR model can be employed for face morphing as an encoder in a GAN with losses specifically adapted to face morphing. Our network gradually morphs two faces depending on a single parameter α yielding remarkable results.

- In our exhaustive analysis, which emphasizes on faces taken in the wild, we demonstrate how the accuracy of an FR system is affected by morphed faces and how the knowledge of the FR system influences the results.

2. Related Work

2.1. Face Manipulation

An early face manipulation approach used multi-scale inputs and supervised the generation with an additional illumination loss [24]. With the success of GANs in realistic image synthesis, a generator with an encoder-decoder structure is leveraged by the majority of the methods [5, 10, 15, 26, 29, 30, 33, 41] to obtain photo-realistic results. To generate a face with the desired attributes, a conditional GAN [28] structure can be employed, in which the generator is provided with additional information and the discriminator also acts as a classifier. *E.g.*, Diamant *et al*. [10] incorporated a beauty score, while He *et al*. [15] used a binary attribute vector to guide the image synthesis in the decoder.

In face swapping or mapping the attributes of one face to the other face, the generator's input comprises two images. For this task, Chen *et al*. [5] disentangled makeup and non-makeup latent vectors to generate a new face containing the makeup of one face with the remaining properties (identity, background, attributes) of the other face. Nirkin *et al*. [30] proposed a face swapping GAN, which contains multiple generators for reenactment, segmentation, inpainting, and blending. In FaceShifter [25], face identity and attribute features are extracted separately and induced into the decoder at different resolutions. The Mask-Guided GAN [41] further trains a mask to control the region where the features are modified. To combine face identity and face attribute features, the latter two approaches [25, 41] employ spatially adaptive normalization (SPADE) [32], which renormalizes feature maps based on a learn transformation from the features. Ngô *et al*. [29] decomposed the face to adjust the head pose, light, and facial expression separately while maintaining identity and background information.

2.2. GAN Inversion

The objective of GAN inversion is to find the most accurate latent vector, which allows a pretrained GAN to recover the input image. Then, by altering the latent vector, the face can be manipulated. Thus, in contrast to the face manipulation methods mentioned above, GAN inversion only trains the encoder in the generator, whereas the decoder is a pretrained GAN.

Recently, multiple approaches for image synthesis based on latent vectors have been proposed, among which the BigGAN [4] and StyleGAN [20, 21] are most popular. In both StyleGAN versions, Karras *et al*. [20, 21] incorporated adaptive instance normalization (AdaIn) [11, 18] – a similar mechanism to SPADE [32] – to introduce the information of the so-called style vectors into the generator at multiple depths. With the employment of attention as first introduced by Self-Attention GAN [43], the realism and variety of the generated images were further improved [4, 7].

Optimization-based GAN inversion methods [1, 2] first select a random initial latent vector, which then is optimized through gradient descend to produce the desired output image. In their analyses, Abdal *et al*. [1, 2] demonstrated many possibilities with impressive results, including even a smooth transition between two face halves of different identities [2]. Multi-Code GAN [12] utilizes N latent codes to generate N intermediate feature maps, which are then combined, weighted by their adaptive channel importance scores, to recover the output image. In order to invert GANs comprising attention mechanism, Daras *et al*. [7] proposed to employ the discriminator's attention layer. After GAN inversion, localized and semantic-aware edits can be performed by disentangling and clustering the semantic objects in activation maps [6] or leveraging SVMs in the latent space [16, 38]. Venkatesh *et al*. [40] applied the GAN inversion technique from Image2StyleGAN [1] to face morphing by averaging the latent vectors.

Learning-based GAN inversion approaches train a separated encoder, which can be applied to all images and thus dispenses with the need of applying backpropagation to obtain the corresponding latent vector for every image. Zhu *et al*. [49] used a domain-guided encoder as a regularizer to preserve the latent vector within the semantic domain of the generator. The StyleGAN Encoder [36] builds an encoder to extract the feature maps of images and subsequently trains a mapping network to transform the feature maps into layer-specific style vectors, which control the image generation in StyleGAN. Based on the same principle, Xu *et al*. [42] introduced a spatial alignment module into the encoder structure to better capture the spatial information from the input image. By incorporating an iterative refinement mechanism, Alaluf *et al*. [3] drew from the iterative manner of optimization-based methods while maintaining efficiency as no backpropagation is performed. Zhang *et al*. [44] trained a ResNet-50 as a backbone to predict the latent vector and then obtain morphed faces by averaging the latent vectors corresponding to both input faces. Despite remarkable high-quality results, their analysis is restricted to frontal faces and not applicable to faces that are taken in the wild.
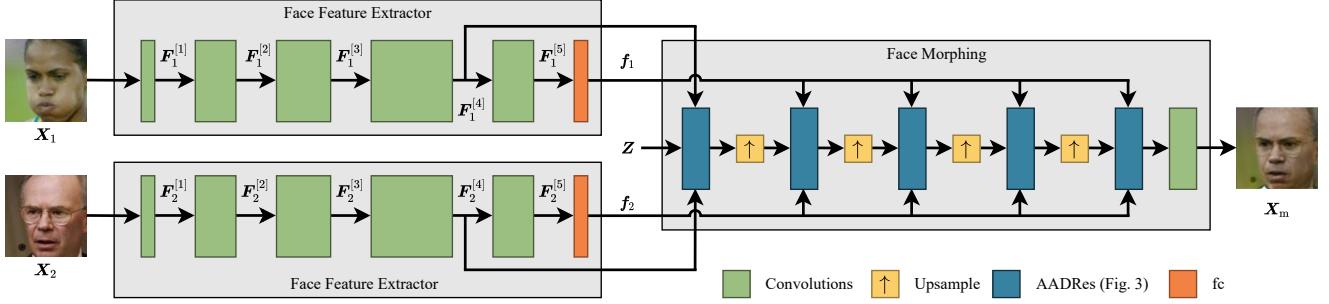
Figure 2. Our approach for face morphing: Face features of two faces $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are extracted by a ResNet-50. Then, the face morphing network utilizes face features $\boldsymbol{f}$ and a single feature map to transform a trainable weight $\boldsymbol{Z}$ into a morphed face $\boldsymbol{X}_{\mathrm{m}}$.

3. Methodology

3.1. Network Architecture

As illustrated in Sec. 2, there are two different architecture choices in training a face morphing network. We opted for a traditional approach not involving GAN inversion. In this way, we can employ a pretrained FR network to encode facial features from which the morphed face $\boldsymbol{X}_{\mathrm{m}}$ is generated. This allows us to investigate how knowledge of the FR system (white-box attack) affects the success rate of a face morphing attack. Due to the task definition, the network must further be invariant to the order of the inputs, *i.e.*, assuming equal weights of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, swapping the inputs must yield the same morphed face $\boldsymbol{X}_{\mathrm{m}}$.

Fig. 2 depicts our approach to face morphing. The features $\boldsymbol{f}$ and intermediate feature maps $\boldsymbol{F}$ of two real faces $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are extracted by a face feature extractor. Then, the morphed face $\boldsymbol{X}_{\mathrm{m}}$ is generated by passing the previously extracted features (maps) through modified adaptive attentional denormalization (AAD) residual blocks [25].

3.1.1 Feature Extractor

The face feature extractor $E_{\mathrm{gen}}(\cdot)$ is used to extract a representation $\boldsymbol{f} \in \mathbb{R}^{256}$ of the face. We use a ResNet-50 [14],
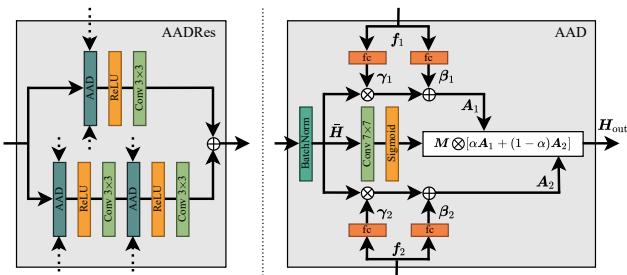

Figure 3. The elements of an adaptive attentional denormalization (AAD) residual block (left) following Li *et al.* [25] together with our modified AAD block (right).

which is pretrained on face identification tasks with softmax cross-entropy loss. Even though additive angular margin (ArcFace [8]) is widely employed nowadays, we decided to use a simpler approach for the face generation in order to demonstrate that even more powerful models, such as those trained with ArcFace, can be deceived by simpler approaches.

3.1.2 Face Morphing

Our face morphing network is inspired by FaceShifter [25], with several adaptions for face morphing. The face morphing network starts with a trainable latent variable $\boldsymbol{Z}$ of size $7 \times 7 \times 512$, which represents the initial feature maps and is deterministic in contrast to StyleGAN [20].

Then, the latent variable $\boldsymbol{Z}$ is propagated through a series of five AAD residual blocks (*cf.* Fig. 3 (left)) with upsampling in between, similar to FaceShifter [25]. While the first AAD residual block is fed with $\boldsymbol{Z}$ and the feature maps $\boldsymbol{F}_1^{[4]}, \boldsymbol{F}_2^{[4]} \in \mathbb{R}^{7 \times 7 \times 1024}$, all subsequent blocks have the upsampled output of the previous blocks and the feature vectors $\boldsymbol{f}_1, \boldsymbol{f}_1$ as inputs. Moreover, the first AAD residual block maintains the number of feature maps and therefore uses a skip connection as the upper path, whereas the remaining AAD residual blocks halve the number of feature maps to reach 32 after the last one. We resize the feature maps with bilinear interpolation as upsampling to reduce checkerboard artifacts, which are frequently introduced by transposed convolutions [31]. Then, the face morphing network is concluded with a 3×3 and 1×1 convolution followed by clipping to obtain the output with the same dimensions and value range as the inputs.

The crucial component of the face morphing network is the AAD block, as illustrated by Fig. 3 (right) for $\boldsymbol{f}$ as input. First, the input feature map $\boldsymbol{H}$ is normalized with a batch normalization layer yielding $\bar{\boldsymbol{H}}$. Based on the normalized input $\bar{\boldsymbol{H}}$, a convolutional layer with sigmoid activation is employed to compute a mask $\boldsymbol{M}$, which indicates the activations in the feature maps to be changed within the AAD

block. Besides the mask prediction, every feature map of the normalized input $\bar{H}$ is de-normalized yielding

$$A = \gamma \bar{H} + \beta, \qquad (1)$$

with the target mean β and variance γ, which are obtained by passing f through a fully connected layer, whose number of neurons match the number of feature maps of $\bar{H}$.

In the first AAD residual block with $F^{[4]}$ as input, the input is flattened before applying the fully connected layer. We found that using $F^{[4]}$ instead of f is crucial to obtain the smooth transition of both images as additional rough spatial information is provided through $F^{[4]}$.

Information about the faces X_1 and X_2 are encoded in A_1 and A_1 as distinct de-normalizations of $\bar{H}$. Unlike in FaceShifter [25], we want to smoothly transition between the faces X_1 and X_2. Hence, we define a scalar parameter $\alpha \in [0;1]$, which globally balances the influence from A_1 and A_2 in every AAD block. Then, the output H_{out} constitutes the element-wise multiplication of the mask M with the balanced encoded face features

$$H_{\text{out}} = M \otimes [\alpha A_1 + (1-\alpha)A_2]. \qquad (2)$$

With our modifications to the original AAD block [25], we obtain invariance with respect to X_1 and X_2 by design. This is achieved by sharing the weights of the face feature extractor and the fully connected layers within every AAD block, which are used to compute γ and β. Moreover, every AAD decides with its own mask M which values of the current feature map to manipulate.

3.2. Loss Functions

To train our face morphing model, we use a weighted sum of several losses

$$\mathscr{L}_{\text{G}} = \lambda_{\text{adv}}\mathscr{L}_{\text{adv}}^{\text{G}} + \lambda_{\text{id}}\mathscr{L}_{\text{id}} + \lambda_{\text{per}}\mathscr{L}_{\text{per}} + \lambda_{\text{style}}\mathscr{L}_{\text{style}}, \qquad (3)$$

where λ_{adv}, λ_{id}, λ_{per}, and λ_{style} denote scalars used to balance the losses.

Similar to most image manipulation approaches in which realism plays a vital role, we utilize the face morphing network as a generator in a GAN structure and train with an adversarial loss. In this way, the face morphing network must generate a photo-realistic face to deceive the discriminator, whereas the discriminator tries to discern real faces X_1 or X_2 from the synthetically generated morphed face X_{m}. We implement a global discriminator $D(X)$ comprising four convolutional layers – the first two with stride two, which are concluded by a fully connected layer and sigmoid activation function denoting the probability of the input image X being real. Then, the adversarial losses are

$$\mathscr{L}_{\text{adv}}^{\text{G}} = -\log(D(X_{\text{m}})), \qquad (4)$$

$$\mathscr{L}_{\text{adv}}^{\text{D}} = -\log(1 - D(X_{\text{m}})) - \frac{1}{2}\sum_{i=1}^{2}\log(D(X_i)). \qquad (5)$$

Similar to [25, 40, 44], we employ an identity loss, which forces the face morphing network to generate the face X_{m} that matches X_1 and X_2. The parameter α indicates how much information the AAD block utilizes from X_1 compared to X_2. Thus, this influence is also reflected in the identity loss

$$\mathscr{L}_{\text{id}} = \alpha d_{\cos}(f_{\text{m}}, f_1) + (1-\alpha)d_{\cos}(f_{\text{m}}, f_2), \qquad (6)$$

where $d_{\cos}(\cdot, \cdot)$ denotes the cosine distance between two feature vectors. To further guide the face morphing network to output a face X_{m} containing information from both inputs, we adapt the perceptual $\mathscr{L}_{\text{per}}$ and style loss $\mathscr{L}_{\text{style}}$ from Johnson *et al.* [19] by incorporating α

$$\mathscr{L}_{\text{per}} = \sum_{i=4}^{5}\frac{\alpha}{N^{[i]}}\left\| F_1^{[i]} - F_{\text{m}}^{[i]} \right\|_1 + \frac{(1-\alpha)}{N^{[i]}}\left\| F_2^{[i]} - F_{\text{m}}^{[i]} \right\|_1,$$
$$(7)$$

with $N^{[i]}$ being the number of elements in $F^{[i]}$. Besides the adversarial loss to ensure photo-realistic results, perceptual loss [19] is widely employed to ensure matching feature maps [5, 10, 12, 26, 29, 30, 36, 42].

The style loss $\mathscr{L}_{\text{style}}$ uses the Gram matrix of every feature map and was modified from [19] accordingly. Both latter losses ensure that the transition of X_{m} from X_1 to X_2 is visible in the feature maps. In contrast to other works on face morphing [40, 44], we decided to compute L_{per} and $\mathscr{L}_{\text{style}}$ based on rather deep feature maps $F^{[4]}$ and $F^{[5]}$ as they do contain less spatial information and thus less ambiguity, *i.e.*, the network is not forced to generate two noses if their corresponding activations are at different locations in shallower feature maps. Moreover, we employ a feature extractor trained on faces to increase the meaningfulness of such feature maps.

4. Experiments

4.1. Training Details

To demonstrate that the morphed face not only deceives the feature extractor used for face morphing $E_{\text{gen}}(\cdot)$, we also utilize a more sophisticated feature extractor trained with ArcFace loss $E_{\text{arc}}(\cdot)$ and apply it on a different dataset. Both feature extractors were trained with facial images of size 112×112, which were aligned utilizing the landmarks obtained by MTCNN [45]. While we use VGGFace2 to train $E_{\text{gen}}(\cdot)$, the refined MS-Celeb-1M was utilized for $E_{\text{arc}}(\cdot)$ [8, 13].

Next, the weights of $E_{\text{gen}}(\cdot)$ are fixed and the face morphing network is trained with $\mathscr{L}_{\text{G}}$ ($\lambda_{\text{adv}} = 1$, $\lambda_{\text{id}} = 2$,

$\lambda_{\text{per}} = 0.5$, and $\lambda_{\text{style}} = 120$) in an alternating manner with the discriminator $\mathcal{L}_{\text{adv}}^{\text{D}}$. To ease convergence, we first train with $\boldsymbol{X}_1 = \boldsymbol{X}_2$ for 5 epochs and finetune with $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$ for another 10 epochs using Adam optimizer. Every batch comprises 32 faces from exactly 16 different identities. We found that the face morphing network improves very slowly and thus only use every fourth batch to train the discriminator. Moreover, the learning rates of the face morphing network and the discriminator are set to 10^{-4} and 10^{-5}, respectively. Both are decayed by a factor of 0.5 every third epoch. For the parameter α, we implement two versions: 1) fixed at $\alpha = 0.5$ throughout the training; and 2) $\alpha = 0.5$ during pretraining and a truncated Gaussian distribution with mean $\mu = 0.5$ and variance $\sigma = 0.2$ for finetuning to ensure that also faces morphed with $\alpha \neq 0.5$ are realistic.

4.2. Benchmark Details

Typical benchmarks for face verification can be seen as a list of triplets $\mathcal{T} = (\boldsymbol{X}_1, \boldsymbol{X}_2, y)$ with $y = 1$ denoting that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ have the same identity ($\text{id}(\boldsymbol{X}_1) = \text{id}(\boldsymbol{X}_2)$) and $y = 0$ if not. For our task, we want to obtain the accuracy Acc_{morph} of a face feature extractor $E_{\text{test}}(\cdot)$ correctly classifying $\boldsymbol{X}_{\text{m}}$ if $\boldsymbol{X}_{\text{m}}$ was generated from two faces with different identities, *i.e.*, the desired classification is $\text{id}(\boldsymbol{X}_{\text{m}}) \neq \text{id}(\boldsymbol{X}_1)$ and $\text{id}(\boldsymbol{X}_{\text{m}}) \neq \text{id}(\boldsymbol{X}_2)$. Formally, Acc_{morph} is computed as

$$Acc_{\text{morph}} = 1 - \frac{1}{N_{\text{diff}}} |\{\forall \, \mathcal{T} \mid d_{\cos}(\boldsymbol{f}_{\text{m}}, \boldsymbol{f}_1) < t \ \& \ y = 0 \ \&$$
$$d_{\cos}(\boldsymbol{f}_{\text{m}}, \boldsymbol{f}_2) < t\}|, \qquad (8)$$

with N_{diff} denoting the number of imposter pairs ($y = 0$) and t the threshold, which is computed to maximize the traditional accuracy of the respective verification protocol. Thus, Acc_{morph} can also be referred to as the failure rate of a face morphing attack onto a FR system. We further compute Eq. (8) for $y = 1$, *i.e.*, genuine pairs, to affirm that the morphed face generated by using two faces from the same identity is perceived as another image of that identity.

Since we have designed our face morphing network to allow a gradual change from $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, $\boldsymbol{X}_{\text{m}}$ always looks similar to both input faces $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Therefore, we extended the triplets in the benchmark to quintuples by adding two images $\boldsymbol{X}_1'$ and $\boldsymbol{X}_2'$, which match the identities of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, respectively.[1] Thus, $\text{id}(\boldsymbol{X}_1) = \text{id}(\boldsymbol{X}_1')$ and $\text{id}(\boldsymbol{X}_2) = \text{id}(\boldsymbol{X}_2')$. Then, $\boldsymbol{X}_{\text{m}}$ is still created based on $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, yet the features of $\boldsymbol{X}_1'$ and $\boldsymbol{X}_2'$ are used for evaluation and to compute the threshold t. This quintuple protocol is denoted by † in our analysis.

For our analysis, we use the LFW [17] dataset together with the cross-age and cross-pose extensions CALFW [48]
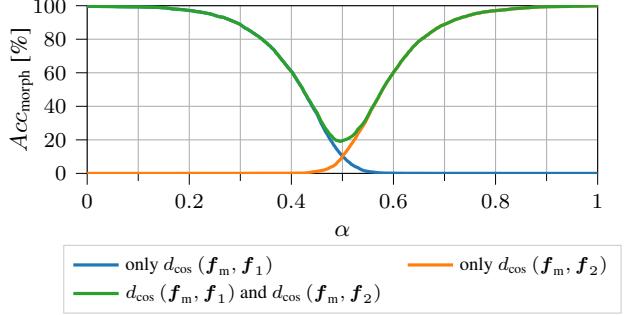
Figure 4. The accuracy Acc_{morph} of ArcFace for different α when distinguishing faces morphed by the model trained with Gaussian-distributed α and without $\mathcal{L}_{\text{style}}$. Faces are morphed based on the LFW benchmark with imposter pairs and Acc_{morph} is computed also separately for every $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

and CPLFW [47]. All three benchmark datasets contain 6000 pairs (3000 imposter and 3000 genuine pairs) and are evaluated using 10-fold cross-validation. Even though nowadays the LFW dataset is not very helpful in evaluating face verification accuracy due to its obvious imposter pairs, it fits our purpose since other datasets ensure that imposter pairs have the same gender and ethnicity, which eases deceiving the FR system. By using $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ to generate $\boldsymbol{X}_{\text{m}}$, we maintain the properties (cross-age and cross-pose) of the CALFW and CPLFW dataset, whereas the threshold t computation is based on $\boldsymbol{X}_1'$ and $\boldsymbol{X}_2'$ to ensure that the same threshold is used to distinguish $\boldsymbol{X}_1'$ from $\boldsymbol{X}_2'$, and $\boldsymbol{X}_{\text{m}}$ from $\boldsymbol{X}_1'$, $\boldsymbol{X}_2'$. However, our method does not guarantee an age or a pose difference between the newly selected $\boldsymbol{X}_1'$ and $\boldsymbol{X}_2'$ as in the original CALFW and CPLFW benchmarks. Still, the same gender and ethnicity are maintained as defined in the original protocols.

The extension from triplets to quintuples requires modifications to the original LFW protocol as many identities in the LFW dataset only have a single image. 1165 genuine and 2736 imposter pairs were replaced, reducing the number of identities covered by the benchmark from 3158 to 1648. For CALFW and CPLFW, no pairs were substituted as at least two images per identity were available. Despite the inherent reduction of generalization due to fewer identities in the quintuples LFW, the frequent differences of ethnicity and gender in imposter pairs still render it particularly interesting.

4.3. Quantitative Results

Fig. 4 illustrates the change in accuracy Acc_{morph} as introduced in Eq. (8) together with the accuracies per identity. It is apparent that our approach provides a smooth transition between two faces in the feature space $\boldsymbol{f}$. While $\alpha \approx 0$ results in a morphed face $\boldsymbol{X}_{\text{m}}$, which is never classified to have the same identity as $\boldsymbol{X}_1$ resulting in an accuracy close to 100%, it contains enough information to be classified as

Table 1. Ablation study: Accuracy Acc_{morph} [%] of the FR system $E_{\text{test}}(\cdot)$ when classifying morphed faces $\boldsymbol{X}_{\text{m}}$ on LFW, CALFW, and CPLFW datasets. *Same* and *diff* denote whether the input images have the same identity and † indicates that different images were used for evaluation than for morphing.

Features for face morphing	$\mathscr{L}_{\text{style}}$	α	LFW $E_{\text{test}}(\cdot)=E_{\text{gen}}(\cdot)$ same	diff	$E_{\text{test}}(\cdot)=E_{\text{arc}}(\cdot)$ same	diff	same †	diff †	CALFW $E_{\text{test}}(\cdot)=E_{\text{arc}}(\cdot)$ diff	diff †	CPLFW diff	diff †
$\boldsymbol{f}$	√	0.5	**0.0**	1.1	0.2	32.2	2.1	72.9	19.0	71.7	19.1	73.9
$\boldsymbol{f}, \boldsymbol{F}^{[3]}, \boldsymbol{F}^{[4]}$	√	0.5	**0.0**	0.5	0.1	19.7	0.8	62.8	7.7	58.9	19.2	71.4
$\boldsymbol{f}, \boldsymbol{F}^{[4]}$	√	0.5	**0.0**	**0.0**	0.1	**18.2**	0.9	62.7	6.4	59.4	**15.3**	68.3
$\boldsymbol{f}, \boldsymbol{F}^{[4]}$		0.5	**0.0**	**0.0**	0.1	19.2	0.9	62.2	6.4	58.5	16.5	**68.2**
$\boldsymbol{f}, \boldsymbol{F}^{[4]}$	√	$\mathcal{N}(0.5, 0.2)$	**0.0**	4.3	**0.0**	19.6	**0.6**	**60.8**	5.2	54.5	16.2	69.7
$\boldsymbol{f}, \boldsymbol{F}^{[4]}$		$\mathcal{N}(0.5, 0.2)$	0.1	0.8	0.1	19.5	0.7	61.1	**4.7**	**54.2**	16.1	68.7

id($\boldsymbol{X}_2$) and fool the system. For $\alpha \approx 1$, this behavior is inverted. The lowest accuracy of rejecting $\boldsymbol{X}_{\text{m}}$ for at least one identity $Acc_{\text{morph}} = 19.5\%$ is achieved for $\alpha \approx 0.5$, where the information from $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ to generate $\boldsymbol{X}_{\text{m}}$ is considered equally. Still, the network classifies $\boldsymbol{X}_{\text{m}}$ as either id($\boldsymbol{X}_1$) or id($\boldsymbol{X}_2$) in 89.8% of the cases.

Tab. 1 depicts the results of our approach for different configurations. When considering genuine pairs (same), it is evident that the morphed face $\boldsymbol{X}_{\text{m}}$ is always classified as the identity – even in the more challenging scenario when the faces used for morphing and evaluation differ (†).

In the more applicable scenario of morphing faces from imposter pairs (diff), the differences between the configurations become apparent. Morphing a face solely based on two 256-dimensional identity vectors $\boldsymbol{f}$ yields inferior results on all protocols. This demonstrates that the spatial information present in $\boldsymbol{F}^{[4]}$ is crucial for achieving satisfying results. Incorporating feature maps $\boldsymbol{F}^{[3]}$ with a resolution of 14×14 leads to worse results. We conjecture that including $\boldsymbol{F}^{[3]}$ confuses the network in many cases as information in $\boldsymbol{F}^{[3]}$ is more ambiguous due to the larger resolution.

According to the results reported in Tab. 1, utilizing style loss $\mathscr{L}_{\text{style}}$ in addition to the perceptual loss $\mathscr{L}_{\text{per}}$ cannot be considered beneficial. Independent of α, not employing $\mathscr{L}_{\text{style}}$ results in a slight improvement, which becomes more noticeable on the more challenging and relevant cases with quintuples (†). Gaussian-distributed α lowers Acc_{morph} most noticeably on CALFW.

The inferior accuracies Acc_{morph} on CALFW compared to LFW confirm the suitability of LFW for this analysis since imposter pairs in CALFW were selected to have the same gender and ethnicity, which facilitates face morphing. Even when morphing faces with large head poses variations as in CPLFW, the face morphing network deceives $E_{\text{arc}}(\cdot)$ with a success rate of over 30%. Still, varying head poses represent one of the biggest challenges in face morphing, which is also affirmed by visual inspection of the feature distances in Fig. 5. When using an operating threshold
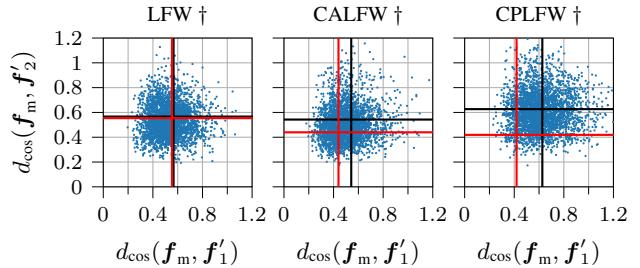

Figure 5. Cosine feature distances between $\boldsymbol{X}_{\text{m}}$, generated by the model trained with Gaussian-distributed α and without $\mathscr{L}_{\text{style}}$, and $\boldsymbol{X}_1'$, $\boldsymbol{X}_2'$ extracted with ArcFace. Operating thresholds maximizing the traditional accuracy (black) or obtaining a false accept rate $FAR = 0.1\%$ (red) indicate the decision boundaries. Thus, distances lying in the bottom left quadrant were misclassified.

typical in security-sensitive applications corresponding to a false accept rate $FAR = 0.1\%$, only 1.3% of the morphed faces can fool the FR system. Still, one must note that this behavior is expected in challenging scenarios and leads to relatively low true accept rates rendering the whole system impractical.

Our analysis also highlights the dependency on the evaluation protocol. When considering a white-box attack, *i.e.*, using the same feature extractor for morphing and evaluation $E_{\text{test}}(\cdot)=E_{\text{gen}}(\cdot)$, the face morphing network learns to fool this specific system limiting Acc_{morph} to $\approx 0\%$ for most models, even if different identities are morphed. However, if a more sophisticated model $E_{\text{arc}}(\cdot)$ is employed for testing, the FR system detects morphed faces with $Acc_{\text{morph}} > 18\%$. This is reasonable as the morphed face was not generated to deceive $E_{\text{arc}}(\cdot)$, which focuses on different face features due to its distinct loss and training dataset. For the most practical and challenging scenario of using different faces for morphing and evaluation (†), the accuracy Acc_{morph} exceeds 60%. Nevertheless, being able to fool the system in 30-45% of the cases, depending on the gender and pose of both faces, clearly demonstrates the susceptibility of
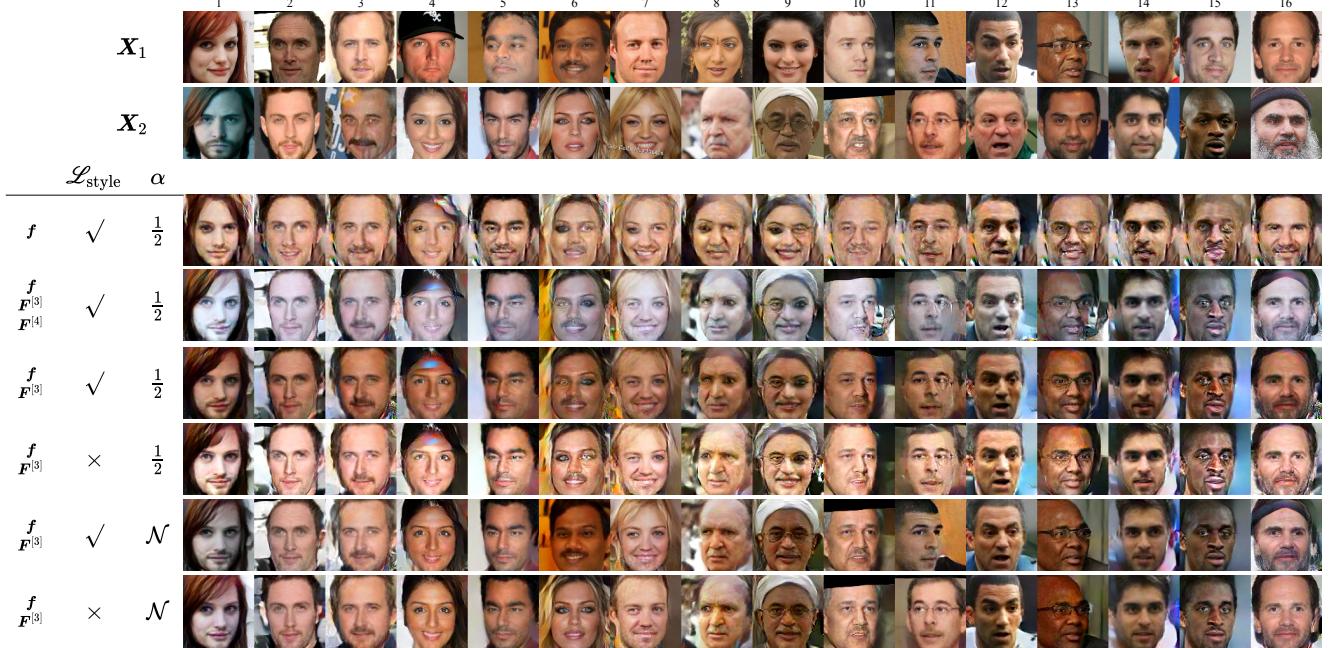
Figure 6. Morphed faces $\boldsymbol{X}_{\mathrm{m}}$ generated from $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by models trained with different parameters following Tab. 1. $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are selected from LFW dataset and $\boldsymbol{X}_{\mathrm{m}}$ is computed for $\alpha = 0.5$.

Figure 7. Gradual change of morphed faces $\boldsymbol{X}_{\mathrm{m}}$ generated by the model trained with Gaussian-distributed α and without $\mathscr{L}_{\mathrm{style}}$.

state-of-the-art methods to face morphing. Besides, if one would want to fool a system using frontal faces of similar ethnicity and gender would be an obvious choice, which results in a success rate of up to 45.8%.

4.4. Qualitative Results

Many morphed faces $\boldsymbol{X}_{\mathrm{m}}$ in Fig. 6 display noticeable artifacts, making it easy for a human to spot that the face must have been manipulated. Still, with the rise of automatic border control or automatic access systems, missing realism is only a small disadvantage if plausibility checks are not employed. Moreover, the last two rows, *i.e.*, employing Gaussian-distributed α during training, showed the most realistic results in accordance with Tab. 1. Particularly interesting results are further shown by the model only provided with features f_1 and f_2. Here, the absence of spatial information causes the model to generate always frontalized $\boldsymbol{X}_{\mathrm{m}}$, which further demonstrates that certain information such as accessories are not encoded into f_1 and f_2 in the first place. Fig. 7 visually confirms the quantitative analysis in Fig. 4 in that our face morphing network achieves a seamless change between two faces.

5. Conclusion

This paper presents a method of using an existing pre-trained FR model to generate morphed faces. The FR model is used to extract face identity features and feature maps, which guide the decoder in generating a morphed face. By adapting the AAD block and multiple losses to face morphing, we achieve a seamless transition between two faces – visually and in the feature space. Compared to previous approaches, we also encompass pairs of faces with varying head poses, different gender, or ethnicity. Our exhaustive analysis demonstrates that state-of-the-art FR are vulnerable to morphed faces even if a relatively simple FR model is employed to generate the morphed face. Besides, we analyze the influence of knowing the FR model (white-box attack) and show that morphed faces with extreme head poses are less likely to be misclassified. Overall, our work highlights the necessity of using deepfake detection – particularly when employing FR in security-sensitive scenarios.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to Edit the Embedded Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 2

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2

[5] Hung-Jen Chen, Ka-Ming Hui, Szu-Yu Wang, Li-Wu Tsao, Hong-Han Shuai, and Wen-Huang Cheng. BeautyGlow: On-Demand Makeup Transfer Framework With Reversible Generative Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 4

[6] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in Style: Uncovering the Local Semantics of GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 2

[7] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your Local GAN: Designing Two Dimensional Local Attention Mechanisms for Generative Models. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 14531–14539, 2020. 2

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 1, 3, 4

[9] Jiankang Deng, Jia Guo, Jing Yang, Alexandros Lattas, and Stefanos Zafeiriou. Variational Prototype Learning for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11906–11915, 2021. 1

[10] Nir Diamant, Dean Zadok, Chaim Baskin, Eli Schwartz, and Alex M Bronstein. Beholder-GAN: Generation and Beautification of Facial Images with Conditioning on Their Beauty Level. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 739–743. IEEE, 2019. 1, 2, 4

[11] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A Learned Representation For Artistic Style. *arXiv preprint arXiv:1610.07629*, 2016. 2

[12] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image Processing Using Multi-Code GAN Prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3012–3021, 2020. 2, 4

[13] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102. Springer, 2016. 4

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity Mappings in Deep Residual Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016. 3

[15] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial Attribute Editing by Only Changing What You Want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019. 1, 2

[16] Stefan Hörmann, Arka Bhowmick, Michael Weiher, Karl Leiss, and Gerhard Rigoll. Face Texture Generation And Identity-Preserving Rectification. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2448–2452. IEEE, 2021. 2

[17] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 1, 5

[18] 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*, pages 1501–1510, 2017. 2

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 4

[20] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks . In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 2, 3

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 8110–8119, 2020. 2

[22] Yonghyun Kim, Wonpyo Park, and Jongju Shin. BroadFace: Looking at Tens of Thousands of People at Once for Face Recognition. In *European Conference on Computer Vision*, pages 536–552. Springer, 2020. 1

[23] Pavel Korshunov and Sébastien Marcel. Vulnerability of Face Recognition to Deep Morphing. In *International Conference on Biometrics for Borders*, 2019. 1

[24] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast Face-swap Using Convolutional Neural Networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 1, 2

[25] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019. 1, 2, 3, 4

[26] Yunfan Liu, Qi Li, Zhenan Sun, and Tieniu Tan. A^{3}GAN: An Attribute-Aware Attentive Generative Adversarial Network for Face Aging. *IEEE Transactions on Information Forensics and Security*, 16:2776–2790, 2021. 1, 2, 4

[27] Puspita Majumdar, Akshay Agarwal, Richa Singh, and Mayank Vatsa. Evading Face Recognition via Partial Tampering of Faces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1

[28] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[29] Minh Ngô, Sezer Karaoğlu, and Theo Gevers. Self-supervised Face Image Manipulation by Conditioning GAN on Face Decomposition. *IEEE Transactions on Multimedia*, 2021. 2, 4

[30] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019. 1, 2, 4

[31] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and Checkerboard Artifacts. *Distill*, 1(10):e3, 2016. 3

[32] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 2

[33] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. DeepFaceLab: Integrated, flexible and extensible face-swapping framework. *arXiv preprint arXiv:2005.05535*, 2020. 1, 2

[34] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in Frequency: Face Forgery Detection by Mining Frequency-aware Clues. In *European Conference on Computer Vision*, pages 86–103. Springer, 2020. 1

[35] Le Qin, Fei Peng, Min Long, Raghavendra Ramachandra, and Christoph Busch. Vulnerabilities of Unattended Face Verification Systems to Facial Components-based Presentation Attacks: An Empirical Study. *ACM Transactions on Privacy and Security*, 25(1):1–28, 2021. 1

[36] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2, 4

[37] Ulrich Scherhag, Christian Rathgeb, Johannes Merkle, Ralph Breithaupt, and Christoph Busch. Face Recognition Systems under Morphing Attacks: A Survey. *IEEE Access*, 7:23012–23026, 2019. 1

[38] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the Latent Space of GANs for Semantic Face Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2

[39] Sushma Venkatesh, Kiran Raja, Raghavendra Ramachandra, and Christoph Busch. On the Influence of Ageing on Face Morph Attacks: Vulnerability and Detection. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020. 1

[40] Sushma Venkatesh, Haoyu Zhang, Raghavendra Ramachandra, Kiran Raja, Naser Damer, and Christoph Busch. Can GAN Generated Morphs Threaten Face Recognition Systems Equally as Landmark Based Morphs? – Vulnerability and Detection. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020. 1, 2, 4

[41] Yi Wei, Zhe Gan, Wenbo Li, Siwei Lyu, Ming-Ching Chang, Lei Zhang, Jianfeng Gao, and Pengchuan Zhang. Mag-GAN: High-Resolution Face Attribute Editing with Mask-Guided Generative Adversarial Network. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[42] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative Hierarchical Features from Synthesizing Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4432–4442, 2021. 2, 4

[43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 2

[44] Haoyu Zhang, Sushma Venkatesh, Raghavendra Ramachandra, Kiran Raja, Naser Damer, and Christoph Busch. MIPGAN – Generating Strong and High Quality Morphing Attacks Using Identity Prior Driven GAN. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(3):365–383, 2021. 1, 2, 4

[45] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 4

[46] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-Attentional Deepfake Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2185–2194, June 2021. 1

[47] Tianyue Zheng and Weihong Deng. Cross-Pose LFW: A Database for Studying Cross-Pose Face Recognition in Un-

constrained Environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5:7, 2018. 5

[48] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. *arXiv preprint arXiv:1708.08197*, 2017. 5

[49] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-Domain GAN Inversion for Real Image Editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 2