Hindawi
IET Biometrics
Volume 2023, Article ID 9353816, 15 pages
https://doi.org/10.1049/2023/9353816

Research Article

Worst-Case Morphs Using Wasserstein ALI and Improved MIPGAN

U. M. Kelly,[1] **M. Nauta**,[1] **L. Liu**,[1] **L. J. Spreeuwers**,[1] **and R. N. J. Veldhuis**[1,2]

[1] *Data Management and Biometrics Group, Faculty of EEMCS, University of Twente, Enschede 7500 AE, Netherlands*
[2] *Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik, Norway*

Correspondence should be addressed to U. M. Kelly; u.m.kelly@utwente.nl

Received 18 May 2023; Revised 5 September 2023; Accepted 12 September 2023; Published 10 November 2023

Academic Editor: Vitomir Struc

A morph is a combination of two separate facial images and contains the identity information of two different people. When used in an identity document, both people can be authenticated by a biometric face recognition (FR) system. Morphs can be generated using either a landmark-based approach or approaches based on deep learning, such as generative adversarial networks (GANs). In a recent paper, we introduced a *worst-case* upper bound on how challenging morphing attacks can be for an FR system. The closer morphs are to this upper bound, the bigger the challenge they pose to FR. We introduced an approach with which it was possible to generate morphs that approximate this upper bound for a known FR system (white box) but not for unknown (black box) FR systems. In this paper, we introduce a morph generation method that can approximate worst-case morphs even when the FR system is not known. A key contribution is that we include the goal of generating difficult morphs *during* training. Our method is based on adversarially learned inference (ALI) and uses concepts from Wasserstein GANs trained with gradient penalty, which were introduced to stabilise the training of GANs. We include these concepts to achieve a similar improvement in training stability and call the resulting method Wasserstein ALI (WALI). We finetune WALI using loss functions designed specifically to improve the ability to manipulate identity information in facial images and show how it can generate morphs that are more challenging for FR systems than landmark- or GAN-based morphs. We also show how our findings can be used to improve MIPGAN, an existing StyleGAN-based morph generator.

1. Introduction

It has been shown that *morphing attacks* pose a significant risk to both face recognition (FR) systems and humans, e.g., border guards [1, 2]. A morph is an image that is created by combining two images of two different people. If it contains sufficient identity information of each person, then FR systems and humans will accept the morph as a match both when it is compared to a different image of the first person and when it is compared with a different image of the second person. This means that two people could share one passport or other identity document and avoid travel restrictions or border controls, e.g., a criminal could travel using the identity document of an accomplice. Some countries intend to stop allowing people to bring their own printed photos for passport applications, e.g., Germany [3]. At the same time, there are still countries that allow applicants to provide their own digital or printed passport photo, e.g., Ireland [4]. Morphed images also pose a challenge in other scenarios since two people could, for example, share a driver's licence, health insurance, public transportation tickets, etc. There are myriad ways to exploit systems, subscriptions, access rights, and more using morphed images.

Generative adversarial networks (GANs) have been shown to successfully generate fake data that matches a real data distribution [5]. Image characteristics such as expression or age (in the case of facial images) can be manipulated by applying changes to latent representations of images, which are vectors in a GAN's latent space. If an inversion were available that maps images to the latent space of a GAN, then this would allow an advantage to be taken of the benefits that GANs provide and allow real data to be manipulated directly.

Mapping two images onto two respective latent vectors and finding an appropriate interpolation between those two vectors would then lead to a GAN-generated morph. Both MorGAN [6] and MIPGAN [7] are examples of this approach.

Morph generation can rely on landmark-based, GAN-based, or manual methods. More recently, morphs generated using diffusion models were introduced [8, 9]. How challenging morphs are varies depending on implementation details such as the landmark detector used, the splicing method used, postprocessing, whether images were printed and scanned, which pairs of images were selected for morphing, etc. A criminal could make a morph using hand-selected landmarks and then iteratively apply changes and test the morph using one or more FR systems to find a morph that is most likely to be accepted by FR systems. They could also apply changes that make it harder for morphing attack detection (MAD) methods to detect the morphs. This means that the variation in morphing methods used in research may not be representative of morphs that could exist in reality, since criminals will not advertise which morphing methods they are using. Therefore, the estimated vulnerability of FR and MAD systems may be different on such morphs than on datasets generated by researchers, where some trade-off between quantity and quality may have to be made.

MAD methods have been proposed, targeted at detecting landmark-based morphs, GAN-based morphs, or both. Developing an MAD approach that can detect both landmark- and GAN-based morphs—especially if they are of a type not seen during training—is still an open challenge [10]. GAN-based morphing detection is very similar to the general detection of GAN images (deepfakes) [11]. Increasing the variation in available morphing tools could be helpful in the development of detection methods, since both in GAN-based morph detection and deepfake detection, more generally, it has been shown that methods struggle to detect images of a type not seen during the training phase.

In a previous study [12], we showed that theoretically—and if the FR system is known also in practice—morphs can be even more challenging than either landmark- or GAN-morphs. While landmark-based morphing combines images in the image domain, GAN-based morphing combines them by mapping them to embeddings in the GAN latent space, interpolating in that latent space, and generating a morph from the interpolated latent embedding. On the other hand, our approach in Kelly et al. [12], was to directly reverse the mapping from images to latent embeddings in the FR latent space (different from the GAN latent space). This approach can be used to exploit the vulnerabilities of the FR system it was trained with but is less suited than GAN-based methods to generate morphs that visually (to humans) look like both contributing identities and struggles to fool unseen FR systems.

In this work, we continue this investigation to find out whether it is possible to automatically generate morphs that approximate the theoretical *worst case* for more than one FR system simultaneously, even when the FR system is unknown ("black box"), showing there are morphs that can be even more challenging than landmark- or GAN-based morphs. The variation of morphs used in existing MAD benchmarks, such as [13–15], can be increased by including approximations of worst-case morphs.

Our contributions consist firstly of adapting the method introduced in adversarially learned inference (ALI) [16] and improving it to better enable manipulation of real data, e.g., generating interpolations of real images. We call the resulting improved method Wasserstein ALI (WALI) and use it to generate morphs. Like ALI, WALI jointly learns a generative and an inverse mapping, enabling its use for morph generation. We improve training stability, which allows the generation of larger images: we generate images using WALI of up to 512 × 512 pixels, compared to 64 × 64 pixels achieved by ALI. It may be possible to generate images with even higher resolutions using WALI, but we did not try this due to hardware and time restraints. ALI's aim is to generate images that look as real as possible, which means it is not necessarily optimal for generating *challenging* morphs. WALI is further improved for this purpose by including loss functions designed specifically to improve the ability to manipulate identity information in facial images. The resulting model provides an easy way to generate (large) morphing datasets intended for training or evaluating FR and MAD systems.

Our second set of contributions lies in applying WALI and our improved implementation of MIPGAN to approximate worst-case morphs, evaluating these approximations, and comparing them to other morphs. Since morphs generated using an underlying StyleGAN Generator [17] are currently the SOTA when it comes to GAN-based morphing, we include MIPGAN morphs in all our comparisons. Summarising, our main contributions are as follows:

(i) Improving ALI to enable morph generation, resulting in WALI, which provides an easy way to generate (large) morphing datasets intended for training or evaluating FR and MAD systems.

(ii) Showing that already considering the goal of generating difficult morphs *during* training instead of only during optimisation (after training) leads to more challenging morphs in both white-box and black-box settings than if WALI is only trained to generate real-looking images.

(iii) Showing that optimisation on our trained model leads to morphs that are more challenging for FR systems than landmark- or MIPGAN-morphs, even when evaluating under black-box settings. This proves the existence of morphs that lie closer (than landmark or MIPGAN) to the theoretical worst-case morph for six out of eight FR systems we evaluated.

(iv) Showing that optimising towards a worst-case embedding is also possible when using existing generative models. Since we see that WALI does not generalise well to new datasets that are different from the data it was trained on, we also apply some of our suggested improvements to a StyleGAN Generator that is better at generalising to new datasets, resulting in an improved MIPGAN approach that also leads to more challenging morphs than other GAN-based approaches.

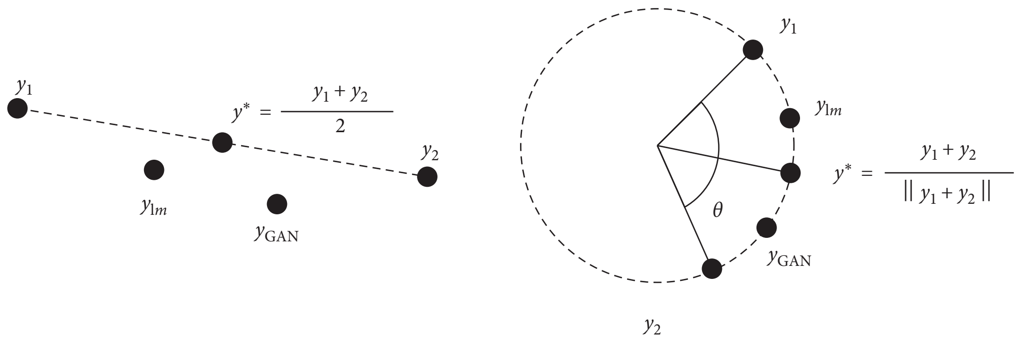

y_1
$y^* = \frac{y_1 + y_2}{2}$
y_2
y_{lm}
y_{GAN}
y_1
y_{lm}
$y^* = \frac{y_1 + y_2}{\| y_1 + y_2 \|}$
θ
y_{GAN}
y_2

Figure 1: The worst-case embedding $\boldsymbol{y}^*$ when d denotes Euclidean distance (left) or angle (right). If it exists, an image that maps to $\boldsymbol{y}^*$ is even more challenging than a landmark- ($\boldsymbol{y}_{\mathrm{lm}}$) or GAN-based morph ($\boldsymbol{y}_{\mathrm{GAN}}$).

2. Related Work

2.1. Worst-Case Morphs. In the study of Kelly et al. [12], an upper bound on the vulnerability of FR systems to morphing attacks was introduced. Let φ be the function that describes an FR system's mapping from the image space X to the embedding space Y, i.e., $\varphi : X \rightarrow Y$. If d is the dissimilarity score function that is used to calculate the dissimilarity score for pairs of embeddings in Y, then the *worst-case embedding* for two images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is

$$\boldsymbol{y}^* := \mathrm{argmin}_{\boldsymbol{y} \in Y}(\max[d(\boldsymbol{y}, \varphi(\boldsymbol{x}_1)), d(\boldsymbol{y}, \varphi(\boldsymbol{x}_2))]). \quad (1)$$

For example, if d returns the euclidean distance, denoted as $\|.\|_2$, between two embeddings $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, then the dissimilarity score is $d(\boldsymbol{y}_1, \boldsymbol{y}_2) = \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2$. In that case $\boldsymbol{y}^*$ is that $\boldsymbol{y}$ for which $d(\boldsymbol{y}_1, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{y}_2) = d(\boldsymbol{y}_1, \boldsymbol{y}_2)/2$, see the example on the left in Figure 1.

If an FR system uses similarity scores, defined by a function S, then

$$\boldsymbol{y}^* := \mathrm{argmax}_{\boldsymbol{y} \in Y}(\min[S(\boldsymbol{y}, \varphi(\boldsymbol{x}_1)), S(\boldsymbol{y}, \varphi(\boldsymbol{x}_2))]). \quad (2)$$

For example, if S returns cosine similarity, then $S(\boldsymbol{y}_1, \boldsymbol{y}_2) = \cos(\theta)$, where θ is the angle between $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, see Figure 1. In that case $\boldsymbol{y}^*$ is any $\boldsymbol{y}$ for which $S(\boldsymbol{y}_1, \boldsymbol{y}) = S(\boldsymbol{y}, \boldsymbol{y}_2) = \cos(\theta/2)$.

Since worst-case embeddings can be calculated using only normal (bona fide) images, no morphs are needed to compute the worst-case upper bound. This means that the potential vulnerability of an FR system can be determined without having to make or evaluate one single morph.

2.2. GANs for Morph Generation. MorGAN [6] uses ALI to generate 64 × 64 pixel morphs. ALI consists of training three networks: an Encoder, a Decoder (similar to the generator in a plain GAN), and a Discriminator. MorGAN generates morphs by passing two images through the encoder, interpolating between the two resulting latent embeddings, and then passing this interpolation through the decoder. This approach results in an image that shares similarities with both original images. Resulting morphs have low resolution and compared to landmark-based morphs are not nearly as successful at fooling FR systems.

MIPGAN [7] makes use of a pretrained StyleGAN network by training an encoder that encodes images into the StyleGAN latent space. Optimisation is then used to approximate an optimal embedding in the StyleGAN latent space, that when passed through StyleGAN results in a morph. The morphs are visually convincing, as confirmed by studies on the human ability to distinguish between morphs and real images. They are about as successful at attacking FR systems as landmark-based morphs. The MIPGAN method is improved on in RegenMorph [18]. The resulting images are visually more convincing but are shown to be less successful than MIPGAN morphs at fooling FR systems.

What these existing GAN-based images have in common is that the underlying networks were all trained with the goal of generating fake images that look like real images. While MorGAN uses a pixel-based loss to preserve identity in images, none of the networks were specifically *trained* to generate morphs. This means that optimisation may be used together with a trained and frozen network to find the optimal latent embedding that leads to a successful morph, but we hypothesise that already considering the goal of generating morphs *during* instead of only *after* training might lead to more successful morphs. Morphing attacks generated specifically to exploit vulnerabilities of deep-learning-based FR can be considered as a type of *adversarial attack* on an FR system [19], since images are manipulated in a way similar to *impersonation attacks*, where in the case of morphing, two identities are being "impersonated" simultaneously.

An overview of research on GAN inversion is provided in the study of Xia et al. [20], where new inverse networks are trained to invert already existing GANs. On the other hand, approaches such as in the study of Dumoulin et al. [16] and Donahue et al. [21] attempt to *jointly* train an encoder, a decoder (the GAN generator), and a discriminator network. As mentioned in the study of Dumoulin et al. [16], it is possible that there are interactions that can be better learned by training these networks jointly, since the Encoder and Decoder can interact during training, which is not possible when using a frozen GAN. For this reason, we explore whether it is possible to improve methods that use the second approach, such as [16, 21], by addressing some disadvantages, such as unstable training. We show that the resulting approach, WALI, is well-suited to approximate worst-case morphs.

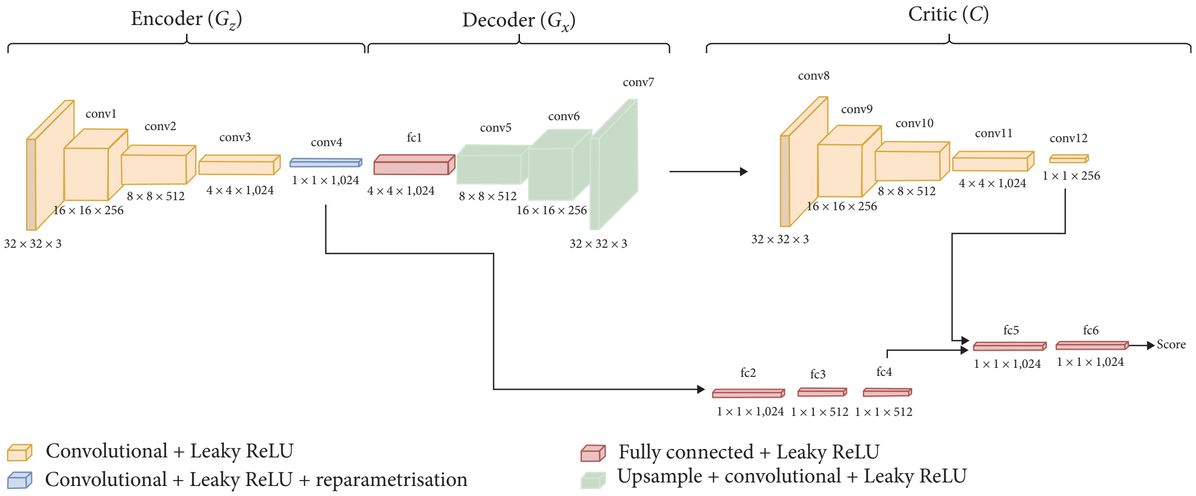


FIGURE 2: The networks and architecture used in Wasserstein ALI (WALI).

2.3. MAD. Research on variation in morphing generation algorithms includes postprocessing landmark-based morphs to mask effects caused by the morphing process [22], a model to simulate the effects of printing and scanning [23], and considering the influence of ageing on morphing attacks [24]. The lack of variation in morphing techniques is addressed in the study of Scherhag et al. [25], which presents a method for MAD and evaluates it on morphs created using different algorithms, which are all landmark-based. Printed-and-scanned morphs are included in this evaluation, but GAN morphs or other methods are not taken into consideration.

In this work, we evaluate morphs using two MAD methods to show that if they are trained with landmark-based morphs only, then they struggle to detect WALI—as well as (improved) MIPGAN-based morphs, emphasising the need for varied datasets for training MAD.

3. Proposed System

3.1. ALI. In ALI [16], two probability distributions over $\boldsymbol{x}$ and $\boldsymbol{z}$ are considered:

(i) the encoder joint distribution $q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})q(\boldsymbol{z}|\boldsymbol{x})$,
(ii) the decoder joint distribution $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$.

The encoder marginal $q(\boldsymbol{x})$ is the empirical data distribution over the image space $\mathscr{X} = [0, 1]^{d_1}$, where $d_1 = w \times h \times n_c$, the width by height of the image by the number of colour channels n_c. The decoder marginal $p(\boldsymbol{z})$ over the latent space $\mathscr{Z}$ is the distribution from which input noise is sampled, e.g., a standard normal distribution $p(\boldsymbol{z}) = \mathcal{N}(0, I)$ over $\mathscr{Z} = (-\infty, \infty)^{d_2}$ (this can be truncated to $[-R, R]^{d_2}$, $R \in \mathbb{R}$ to ensure that $\mathscr{Z}$ is compact, which is needed to prove that ALI converges). Embeddings in the ALI latent space $\mathscr{Z}$ are denoted $\boldsymbol{z}$ and should not be confused with embeddings $\boldsymbol{y}$ in the FR latent space.

The objective of ALI is to match the two joint distributions. In order to achieve this, an adversarial game is played using the following:

(i) G_z: an encoder that maps from image space to a latent space,
(ii) G_x: a decoder that maps from the latent space to image space,
(iii) D (or C): a discriminator (or critic) that tries to determine whether joint pairs $(\boldsymbol{x}, \boldsymbol{z})$ are drawn either from $q(\boldsymbol{x}, \boldsymbol{z})$ or $p(\boldsymbol{x}, \boldsymbol{z})$.

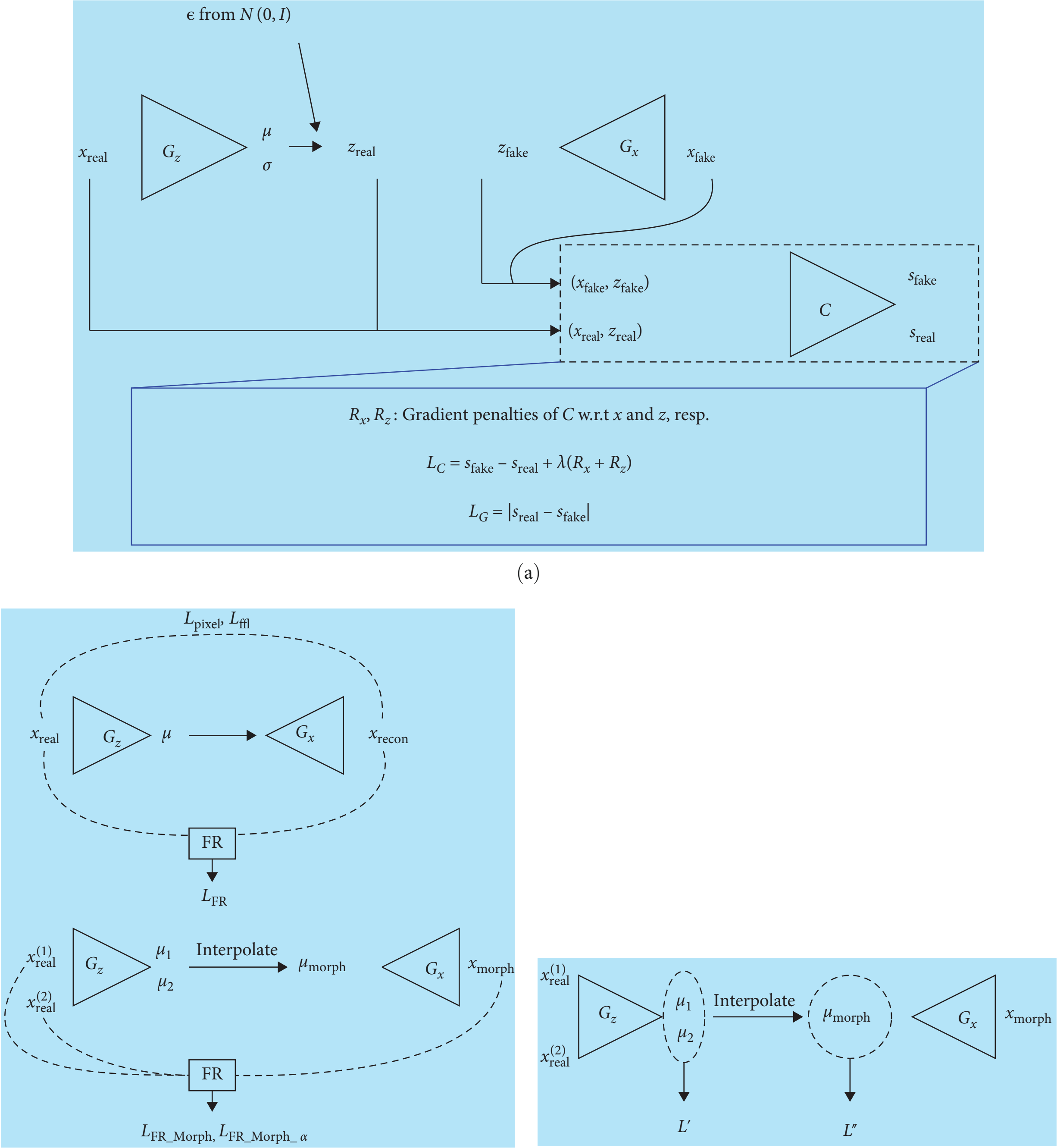
See Figure 2 for a visualisation of these networks.

If the two joint distributions are successfully matched, then existing data points can be encoded into latent vectors that follow the same distribution as the sampled input noise. Then, if the latent vectors are passed through the decoder, the generated images, in turn, follow the same distribution as the real images. These properties together allow us to manipulate existing data and to interpolate between real data points.

ALI suffers from some limitations, such as training instability and limited ability to faithfully reconstruct images [26, 27]. We find that to successfully train ALI to generate facial images, some tweaks are needed, such as limiting the updates of the discriminator and ending training before mode collapse occurs. For this reason, we combine the advantages of Wasserstein GANs [28, 29] with the ALI architecture to improve training stability.

First, we adapt ALI to include Wasserstein elements and train until convergence, see Figure 3(a). Next, we finetune using losses to encourage the system to generate difficult morphs. We do this using losses on the image level that encourage the system to faithfully reconstruct normal images, but also use an FR system to ensure the reconstructed images maintain identity information, see Figure 3(b). We use the same FR system to nudge the system to generate morphs that approximate worst-case morphs, see Figure 3(c).

3.2. Baseline Training. We mainly follow the ALI training procedure but replace transposed convolutions with upsampling and size-maintaining convolutions to avoid chequerboard artefacts [30]. We also remove the sigmoid output layer in the discriminator so that it no longer outputs values between 0



(a)

(b)

(c)

Figure 3: The losses used in WALI: (a) baseline losses, see Section 3.2; (b) losses for finetuning, Section 3.3; (c) losses from Section 3.4 are used to optimise the selection of latent embeddings.

and 1 (where 0 is fake and 1 is real) but instead outputs a *score*, making the discriminator network a *Critic* (C). A higher critic score indicates that an image looks more real, and a lower score indicates that, according to the critic, the generated image looks more fake. We follow the approach from Gulrajani et al. [29], i.e., we update G_z and G_x after every fifth update of the critic. The critic, in turn, is trained to output larger scores for real images and vice versa and to ensure Lipschitz continuity, a gradient penalty is added to the critic loss. Since WALI is trained to match a joint distribution, we include a gradient penalty R_z w.r.t. latent input and a gradient penalty R_x w.r.t. image input. Following recommendations from Gulrajani et al. [29], we set the gradient penalty weight to 10.

We start with a baseline architecture that generates 32 × 32 pixel images. The architecture can be changed to generate higher-resolution images by simply adding layers to the three networks. For example, to generate 64-by-64 pixel images, we add one more convolutional layer before the first layer in G_z and C, and one more upsampling and convolution after the last layer in G_x.

We train C to minimise:

$$\mathscr{L}_C = s_{\text{fake}} - s_{\text{real}} + \lambda(R_x + R_z), \tag{3}$$

and G_z and G_x to minimise:

$$\mathscr{L}_G = |s_{\text{real}} - s_{\text{fake}}|, \tag{4}$$

where

$$s_{\text{real}} = \mathop{\mathbb{E}}_{(\boldsymbol{x}_{\text{real}},\boldsymbol{z}_{\text{real}})\sim q(x,z)} [C(\boldsymbol{x}_{\text{real}}, \boldsymbol{z}_{\text{real}})], \tag{5}$$

$$s_{\text{fake}} = \mathop{\mathbb{E}}_{(\boldsymbol{x}_{\text{fake}},\boldsymbol{z}_{\text{fake}})\sim p(x,z)} [C(\boldsymbol{x}_{\text{fake}}, \boldsymbol{z}_{\text{fake}})], \tag{6}$$

$$R_x = \mathop{\mathbb{E}}_{(\widetilde{\boldsymbol{x}},\widetilde{\boldsymbol{z}})\sim\widetilde{p}(x,z)} \left[\left(\left\|\nabla_{\widetilde{\boldsymbol{x}}} C(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{z}})\right\|_2 - 1\right)^2\right], \tag{7}$$

$$R_z = \mathop{\mathbb{E}}_{(\widetilde{\boldsymbol{x}},\widetilde{\boldsymbol{z}})\sim\widetilde{p}(x,z)} \left[\left(\left\|\nabla_{\widetilde{\boldsymbol{z}}} C(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{z}})\right\|_2 - 1\right)^2\right]. \tag{8}$$

3.3. Finetuning for Morph Generation. Once the three networks G_z, G_x, and C have converged, we fine-tune them using losses that encourage the network to generate morphs that are close to the worst case. We do this using five different losses. The first two losses are a pixel loss $\mathscr{L}_{\text{pixel}}$ and a focal frequency loss (FFL) [31], both encourage the generator network to reconstruct images on a pixel level. This second loss $\mathscr{L}_{\text{ffl}}$ has the advantage that it forces the generator to reconstruct more challenging frequencies as well as easier frequencies.

Next, we define losses to manipulate identity information in generated images using an FR system. Without loss of generality (the only requirement is that d is known and differentiable), we assume the FR system used compares images using a dissimilarity score function d that calculates the angle between two latent embedding vectors. We denote the mapping used by the FR system to map images onto latent embeddings by φ. We use three losses to encourage WALI to generate morphs that contain as much as possible relevant identity information. These are $\mathscr{L}_{\text{FR}}$, $\mathscr{L}_{\text{FR\_Morph\_}\alpha}$, and $\mathscr{L}_{\text{FR\_Morph}}$, which are defined as follows:

$$\mathscr{L}_{\text{FR}} = \mathop{\mathbb{E}}_{\boldsymbol{x}_{\text{real}}\sim q(\boldsymbol{x})} [d(\varphi(\boldsymbol{x}_{\text{recon}}), \varphi(\boldsymbol{x}_{\text{real}}))], \tag{9}$$

where $\boldsymbol{x}_{\text{recon}} = G_x(G_z(\boldsymbol{x}_{\text{real}}))$.

$$\mathscr{L}_{\text{FR\_Morph\_}\alpha} = \mathop{\mathbb{E}}_{(\boldsymbol{x}_{\text{real}},\boldsymbol{z}_{\text{real}})\sim q(x,z)} \left[d\left(\varphi\left(\boldsymbol{x}^{\alpha}_{\text{morph}}\right), \boldsymbol{y}^*\right)\right], \tag{10}$$

where

$$\boldsymbol{x}^{\alpha}_{\text{morph}} = G_x(\alpha \boldsymbol{z}_1 + (1-\alpha)\boldsymbol{z}_2) \tag{11}$$

for $\boldsymbol{z}_1 = G_z(\boldsymbol{x}_1)$ and $\boldsymbol{z}_2 = G_z(\boldsymbol{x}_2)$, and $0 \le \alpha \le 1$. As defined in Equation (1), $\boldsymbol{y}^*$ is the worst-case embedding given $\boldsymbol{y}_1 = \varphi(\boldsymbol{x}_1)$ and $\boldsymbol{y}_2 = \varphi(\boldsymbol{x}_2)$. Finally

$$\mathscr{L}_{\text{FR\_Morph}} = \mathscr{L}_{\text{FR\_Morph\_}\alpha},\ \alpha = 0.5. \tag{12}$$

In principle, $\mathscr{L}_{\text{FR}}$ and $\mathscr{L}_{\text{FR\_Morph}}$ are the same as $\mathscr{L}_{\text{FR\_Morph\_}\alpha}$, just for fixed $\alpha = 1$ and $\alpha = 0.5$, resp. We find that including these losses specifically instead of simply increasing the weight for $\mathscr{L}_{\text{FR\_Morph\_}\alpha}$ leads to the network being able to generate more challenging morphs when evaluating with FR systems under black-box settings, see Table 1.

All five losses are combined in the following Loss:

$$\mathscr{L} = \gamma_1 \mathscr{L}_{\text{pixel}} + \gamma_2 \mathscr{L}_{\text{ffl}} + \gamma_3 \mathscr{L}_{\text{FR}} + \gamma_4 \mathscr{L}_{\text{FR\_Morph}} + \gamma_5 \mathscr{L}_{\text{FR\_Morph\_}\alpha}. \tag{13}$$

We use MobileFaceNet (MFN) [32] to estimate these losses during training, where we intentionally choose a light-weight network in order to reduce the GPU memory needed. We evaluate our generated morphs with seven FR systems that were not used during training: VGG16 [33], ArcFace (AF) [34], the Inception ResNet-based FaceNet (INC) [35], ElasticFace (EF) [36], CurricularFace (CF) [37], PocketNetS-128 (PN) [38], Dlib [39], and a Commercial Off The Shelf (COTS) system.

3.4. Optimisation. After training the three networks, we freeze their weights and optimise the selection of latent embeddings. For each pair of images for morphing, we apply optimisation to select good initial embeddings and then optimise again to find an embedding that when passed through G_x leads to a morph that is close to the worst case. We use an Adam optimiser [40] with hyperparameters $\alpha = 0.05, \beta_1 = 0.9, \beta_2 = 0.999$.

3.4.1. Optimisation Phase 1. Start with $\boldsymbol{z}_1 = G_z(\boldsymbol{x})$ and optimise $\boldsymbol{z}_1$ using the following Loss function:

$$\mathscr{L}' = \|\boldsymbol{x}_1 - G_x(\boldsymbol{z}_1)\|^2 + \|\varphi(\boldsymbol{x}_1) - \varphi(G_x(\boldsymbol{z}_1))\|^2. \tag{14}$$

Do the same to optimise the selection of $\boldsymbol{z}_2$.

3.4.2. Optimisation Phase 2. Start with $\boldsymbol{z}_{\text{morph}} = (\boldsymbol{z}_1 + \boldsymbol{z}_2)/2$ and optimise using the following Loss function:

$$\mathscr{L}'' = \left\|\boldsymbol{y}^* - \varphi\left(G_x\left(\boldsymbol{z}_{\text{morph}}\right)\right)\right\|^2. \tag{15}$$

In both phases, a second FR system can be included to improve the effects of optimisation. Let φ_2 be the mapping corresponding to the second FR system. Then Equations (14) and (15) are extended by adding $\|\varphi_2(\boldsymbol{x}_1) - \varphi_2(G_x(\boldsymbol{z}_1))\|^2$ to $\mathscr{L}'$ and $\left\|\boldsymbol{y}^{**} - \varphi_2\left(G_x\left(\boldsymbol{z}_{\text{morph}}\right)\right)\right\|^2$ to $\mathscr{L}''$, where $\boldsymbol{y}^{**}$ is the worst-case embedding in the second FR system's latent

```
$\theta_D, \theta_{G_z}, \theta_{G_x} \leftarrow$ initialise network parameters
repeat
    $\boldsymbol{x}_{\text{real}}^{(1)}, \dots, \boldsymbol{x}_{\text{real}}^{(N)}$    ▷ Draw $N$ samples from the dataset
    $\boldsymbol{z}_{\text{fake}}^{(1)}, \dots, \boldsymbol{z}_{\text{fake}}^{(N)}$    ▷ Draw $N$ random latent emb.
    $\boldsymbol{z}_{\text{real}}^{(i)} = G_z\left(\boldsymbol{x}^{(i)}\right), \quad i = 1, .., N$    ▷ Get real embeddings
    $\boldsymbol{x}_{\text{fake}}^{(i)} = G_x\left(\boldsymbol{z}_{\text{fake}}^{(i)}\right), \ i = 1, .., N$    ▷ Generate fake images
    $s_{\text{real}} = \frac{1}{N}\sum_{i=0}^{N} C\left(\boldsymbol{x}_{\text{real}}^{(i)}, \boldsymbol{z}_{\text{real}}^{(i)}\right)$    ▷ Critic output for real data
    $s_{\text{fake}} = \frac{1}{N}\sum_{i=0}^{N} C\left(\boldsymbol{x}_{\text{fake}}^{(i)}, \boldsymbol{z}_{\text{fake}}^{(i)}\right)$    ▷ Critic output for fake data
    $\boldsymbol{x}_{\text{recon}}^{(i)} = G_x\left(\boldsymbol{z}_{\text{real}}^{(i)}\right), \quad i = 1, .., N$    ▷ Reconstruct real images
    $\boldsymbol{z}_{\alpha, \text{morph}}^{(i)} = \alpha \boldsymbol{z}_{\text{real}}^{(i)} + (1 - \alpha)\boldsymbol{z}_{\text{real}}^{(j)}, j = 2, .., N, 1$
    $\boldsymbol{z}_{\text{morph}}^{(i)} = \frac{1}{2}\boldsymbol{z}_{\text{real}}^{(i)} + \frac{1}{2}\boldsymbol{z}_{\text{real}}^{(j)}, j = 2, .., N, 1$    ▷ Translate batch to get pairs for morphing
    $\boldsymbol{x}_{\alpha, \text{morph}}^{(i)} = G_x\left(\boldsymbol{z}_{\alpha, \text{morph}}^{(i)}\right), \quad i = 1, .., N$
    $\boldsymbol{x}_{\text{morph}}^{(i)} = G_x\left(\boldsymbol{z}_{\text{morph}}^{(i)}\right), \quad i = 1, .., N$    ▷ Generate morphs
    $\boldsymbol{y}^{(i)} = \varphi\left(\boldsymbol{x}^{(i)}\right), \quad i = 1, .., N$    ▷ Get FR embeddings
    $\boldsymbol{y}_{\alpha}^{*(i)} = \frac{\alpha \boldsymbol{y}^{(i)} + (1-\alpha)\boldsymbol{y}^{(j)}}{||\alpha \boldsymbol{y}^{(i)} + (1-\alpha)\boldsymbol{y}^{(j)}||}, j = 2, .., N, 1$
    $\boldsymbol{y}^{*(i)} = \frac{\boldsymbol{y}^{(i)} + \boldsymbol{y}^{(j)}}{||\boldsymbol{y}^{(i)} + \boldsymbol{y}^{(j)}||}, j = 2, .., N, 1$    ▷ Get worst-case emb.
    $\mathscr{L}_{\text{pixel}} = \frac{1}{N}\sum_{i=0}^{N} \text{MSE}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}_{\text{recon}}^{(i)}\right)$    ▷ Compute pixel loss
    $\mathscr{L}_{\text{ffl}} = \frac{1}{N}\sum_{i=0}^{N} \text{FFL}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}_{\text{recon}}^{(i)}\right)$    ▷ Compute FFL loss
    $\mathscr{L}_{\text{FR}} = \frac{1}{N}\sum_{i=0}^{N}\left(\boldsymbol{y}^{(i)}, \boldsymbol{y}_{\text{recon}}^{(i)}\right)$
    $\mathscr{L}_{\text{FR\_Morph\_}\alpha} = \frac{1}{N}\sum_{i=0}^{N}\left(\boldsymbol{y}_{\alpha, \text{morph}}^{(i)}, \boldsymbol{y}_{\alpha}^{*(i)}\right)$
    $\mathscr{L}_{\text{FR\_Morph\_}} = \frac{1}{N}\sum_{i=0}^{N}\left(\boldsymbol{y}_{\text{morph}}^{(i)}, \boldsymbol{y}^{*(i)}\right)$
                                                        ▷ Compute FR-based losses
    $\mathscr{L}_C = s_{\text{fake}} - s_{\text{real}} + \lambda(R_x + R_z)$    ▷ Compute critic loss
    $\mathscr{L}_G = |s_{\text{fake}} - s_{\text{real}}| + \gamma_1 \mathscr{L}_{\text{pixel}} + \gamma_2 \mathscr{L}_{\text{ffl}} + \gamma_3 \mathscr{L}_{\text{FR}}$
                $+ \gamma_4 \mathscr{L}_{\text{FR\_Morph\_}\alpha} + \gamma_5 \mathscr{L}_{\text{FR\_Morph\_}\alpha}$
                                                        ▷ Compute generator loss
    $\theta_C \leftarrow \theta_C - \nabla_{\theta_C} \mathscr{L}_C$    ▷ Gradient update on critic
    $\theta_{G_z} \leftarrow \theta_{G_z} - \nabla_{\theta_{G_z}} \mathscr{L}_G$    ▷ Gradient update on encoder
    $\theta_{G_x} \leftarrow \theta_{G_x} - \nabla_{\theta_{G_x}} \mathscr{L}_G$    ▷ Gradient update on decoder
until convergence
```

Algorithm 1: Our training procedure.

space. In both phases, the second FR system can be given more or less weight by weighting the new summands.

3.5. Optimisation with Pretrained Generator. All three WALI networks can be trained simultaneously and from scratch. Alternatively, with very little adaptation, our two-phase optimisation approach also allows the use of an existing encoder and generator. We apply optimisation guided by MFN and EF in two phases, similar to optimisation Phases 1 and 2 when using an existing StyleGAN generator and an encoder provided by Tov et al. [41]. This encoder was trained to invert the StyleGAN generator mapping, but unlike WALI, the generator and encoder were not trained simultaneously. Since this approach is similar to MIPGAN [7], we call it *improved MIPGAN*.

4. Experiments

We report results for experiments with WALI models that generate 128 × 128 images, since training time and GPU memory requirements increase significantly when (1) including FR losses during training and (2) increasing the size of the model to generate higher-resolution images. We report our results on 128 × 128 images but have successfully managed to generate visually convincing images up to dimensions of $d_1 = 512 \times 512 \times 3$ (compared to $d_1 = 64 \times 64 \times 3$ for ALI images), see Figure 4.

We train the three WALI networks without any losses other than the Wasserstein and gradient penalty loss until they converge, which takes about 400 epochs with a batch size of 32. We then finetune the networks by adding the losses in Equation (13) and training for another 85 epochs. Our model was implemented with Pytorch [42], training and testing experiments were conducted on a computer equipped with two Nvidia GeForce Titan-X GPUs (12GB).

We use MobileFaceNet [43] to implement the loss functions in Equation (13) during training. To guide optimisation towards an embedding that is "close" to the worst case, we also use MobileFaceNet. Additionally, we report results for experiments in which we used two FR systems during optimisation. We also apply our two-phase optimisation approach using a StyleGAN Generator and an Encoder network from [41], which we call "improved MIPGAN."

4.1. Datasets. We use a dataset of 21,772 facial images from the FRGC dataset [44] and separate them into 18,143 training and 3,629 validation images, with no overlap in identities. We added 32,869 images with frontal pose from FFHQ [17] to the training set. Training without including FFHQ images in the training set was also successful, but including FFHQ improves the results, especially when evaluating with FR (as opposed to only by visual inspection).

We create four sets of morphs using the validation set: landmark-based morphs, GAN-based morphs using MIPGAN-I [7], approximations of worst-case morphs generated by our WALI method, and approximations of worst-case morphs generated using our improved MIPGAN implementation. We select 75 pairs of similar identities by calculating a mean FR embedding for each identity: $\bar{z} = \frac{1}{n}\sum_i^n \varphi(x_i)$, and then selecting those pairs for which the mean FR embeddings are most similar. For each pair of identities, we select all faces with neutral expressions, and from all possible combinations, we randomly select 506 image pairs for morphing.

For each pair (x_1, x_2), we create three landmark morphs, one MIPGAN morph, one WALI worst-case approximation for each FR system used for optimisation (seven in total), and one improved MIPGAN worst-case approximation, see Figure 5. The three landmark morphs comprise one full morph - the faces and also the background of both original images are morphed—and two spliced morphs—full morphs spliced into the

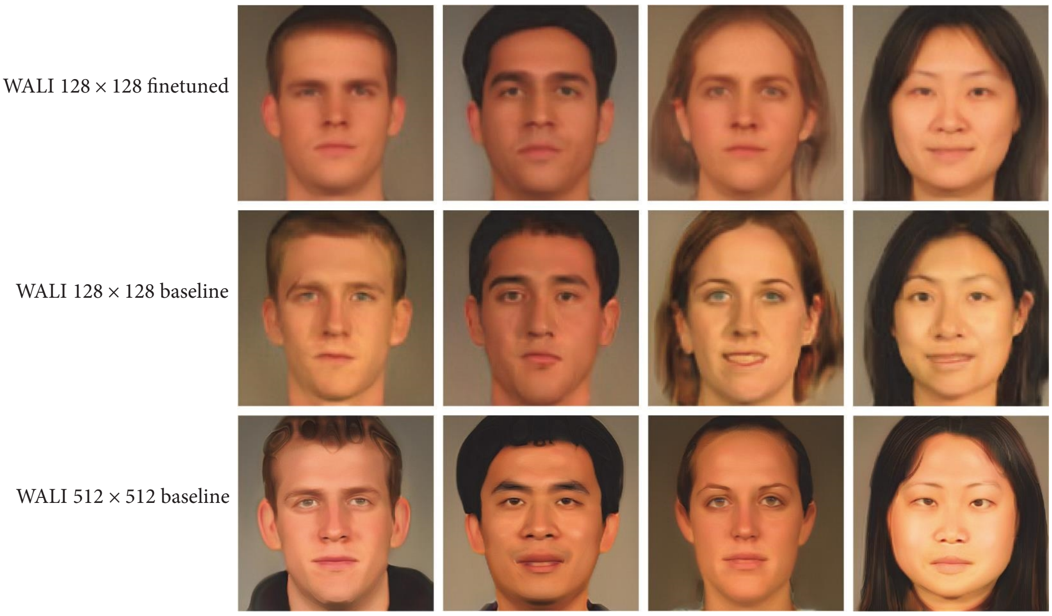


Figure 4: Comparison of morphs generated by a 128 × 128 WALI model trained with FR losses (top row), a 128 × 128 WALI model trained without FR losses (second row) and a 512 × 512 WALI model without FR losses (bottom row). Comparing the top two rows shows that there seems to be a trade-off between image quality and morphing performance. Comparing the two bottom rows shows that blurriness can be corrected by simply generating higher-resolution images.

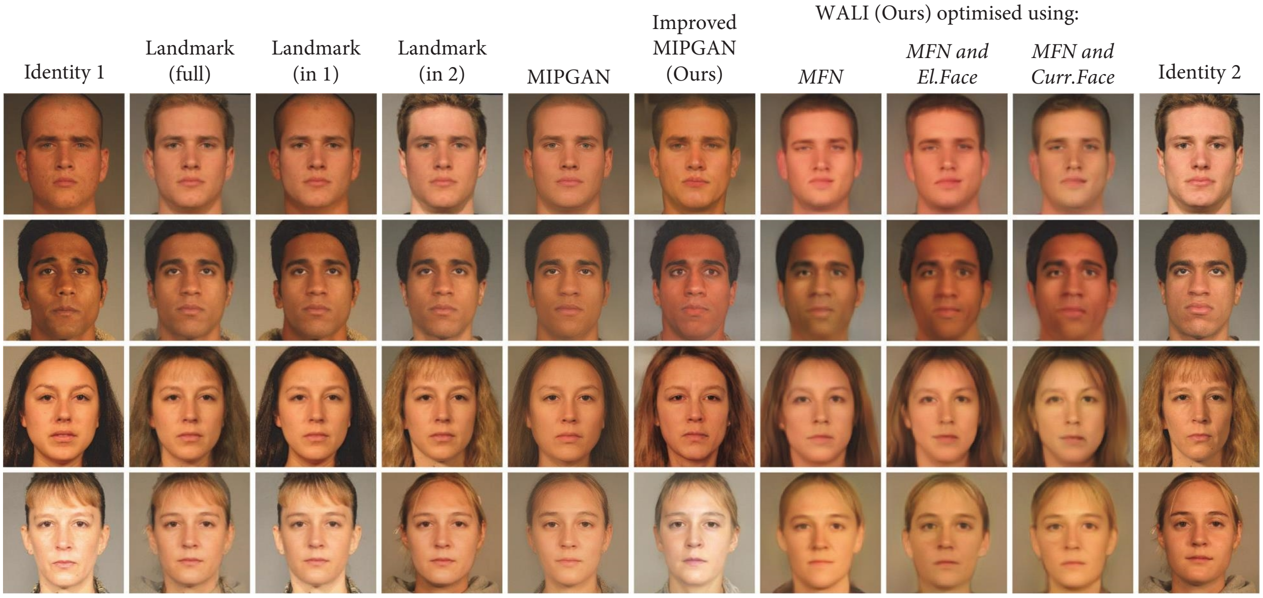


Figure 5: Examples of landmark-based and different GAN-based morphs based on FRGC images.

background of each of the original images, respectively, to remove ghosting artefacts. After freezing WALI's weights, a worst-case approximation is generated by applying 150 optimisation steps in Phase 1 and 150 steps in Phase 2 (Section 3.4). For our improved MIPGAN morphs, we also apply 150 optimisation steps in two phases, this time using a StyleGAN Generator and Encoder. We notice that there is a difference in behaviour between the newer FR systems ElasticFace and CurricularFace compared to the other FR systems we use for optimisation, which we describe in Section 6. For this reason, whenever we optimise with MobileFaceNet and one of these two FR systems, we weight the losses corresponding to the latter with 2. In all other cases, the optimisation losses, as defined in Section 3.4, are weighted equally. We did not extensively analyse the effect of weighting losses differently, so in other applications, this may need to be examined further in order to select weights that suitably balance the different losses. For image generation tools and/or MAD methods, we are aware that it would be better to use datasets that are more balanced and include more variation in terms of gender, age, and ethnicity, and we encourage the research community to take this into consideration for future research.

We also compare different GAN- and landmark-based morphs [45, 46] created using images from the FRLL [47] dataset. The FRLL dataset consists of 102 identities and two images per identity. For each identity, one image with a neutral expression is provided that is suitable for morph generation. Five morph datasets are provided: WebMorph, OpenCV, FaceMorpher, and AMSL are landmark-based morphs, StyleGAN morphs are GAN-based morphs. AMSL consists of 2175 landmark-based morphs. When using a landmark-based morphing tool, a morph based on two images can be spliced into the background of either the first or the second image. Since in the AMSL dataset, both options are not always provided; we only evaluate with identity pairs for which both spliced morphs are provided. We do this to enable a fair comparison of all morphing methods. WebMorph, OpenCV, and FaceMorpher consist of full morphs only, i.e., they contain obvious morphing artefacts.

5. Evaluation Metrics

To measure and compare the performance of our model, we calculate the morphing attack potential (MAP) [48] for $r = 1$ verification attempt and $c = 1$ FR system, which is the same as the mated morph presentation match rate (MMPMR(t)). We consider morphs based on two identities, in which case the MMPMR(t) [1] is the proportion of (morphing) attacks for which both contributing identities are considered a match by the FR system when using a threshold t.

$$\mathrm{MMPMR}(t) = \frac{1}{M} \sum_{m=1}^{M} \{\max(d_1, d_2) < t\}, \tag{16}$$

where d_1 and d_2 are the dissimilarity scores between the morph and a probe image of the first and second identity, respectively. M is the number of morphed images.

We report MMPMR values for nine different FR systems. For each FR system, we set t such that the false nonmatch rate is minimal while the false match rate $<0.1\%$. Higher MMPMR values indicate higher vulnerability to morphing attacks. It would be possible to compute MAP values for $c > 1$, but for WALI morphs, we always treat one or two FR systems as white-box systems, so this might lead to unfair comparisons. Instead, we would have to compute different MAP matrices for all morphing techniques, excluding one or two FR systems at a time, which would become very messy. For this reason, we choose to only report the MMPMR.

5.1. MAD. We evaluate morphs generated using WALI with two MAD methods. The first is a single image-based morphing (S-MAD) approach, based on a support vector machine (SVM) trained with local binary pattern (LBP) features, that learns to detect morphed images based on image texture described using LBP features [49, 50]. The second is a differential image-based (D-MAD) method that is based on deep learning features [51].

We train both MAD methods using the FRGC images we also used to train WALI. While the LBP-based approach can successfully detect WALI morphs, this may simply be due to the similarity of WALI morphs to the FRGC training data. To show that this is the case and that it is insufficient to train with landmark-based morphs only, we also train the LBP approach using FRLL and AMSL. We include 20% of the landmark-based morphs (selected randomly) in the training set due to the class imbalance. Because of the low number of genuine pairs (only one pair per identity), we do not train the D-MAD approach with this dataset.

We report the performance of these two MAD methods using the Bona fide presentation classification error rate (BPCER): the proportion of bona fide images that are incorrectly labelled as morphs, and the attack presentation classification error rate (APCER): the proportion of (morphing) attacks that are misidentified as bona fides. Higher values of BPCER and APCER indicate higher vulnerability of an MAD system to morphing attacks.

6. Results

In Figure 5, we show examples of morphs generated using WALI and compare them with landmark, MIPGAN, and improved MIPGAN morphs. WALI morphs are more blurry compared to MIPGAN morphs, which to a large extent is due to MIPGAN relying on a StyleGAN model that generates 1024×1024 images while the WALI morphs are 128×128-pixel images. In Figure 4, we show that the visual quality (from a human perspective) can be improved simply by increasing the WALI model size.

We report MMPMR values for one FR system at a time for the case where optimisation was guided by MFN only and for the case where optimisation was guided by two FR systems (MFN + EF, MFN + CF, MFN + AF, MFN + INC, MFN + PN, MFN + VGG), see Table 2. Dlib is not available as a Pytorch implementation, so we did not optimise using this FR system. When WALI is optimised with two FR systems, the resulting morphs are more challenging than either landmark or MIPGAN morphs for both FR systems used for optimisation. There is an interesting difference in behaviour that sets apart ElasticFace and CurricularFace from other FR systems. Comparing WALI morphs optimised with MFN + AF, MFN + INC, MFN + PN, MFN + VGG to landmark- and MIPGAN-morphs, we see that the MMPMR is closer to the worst case for all black-box tested FR systems except ElasticFace, CurricularFace, Dlib, and COTS. At first glance, this could be interpreted to mean that ElasticFace and CurricularFace are generally less vulnerable to GAN-based morphing attacks. However, when WALI morphs are optimised using ElasticFace, the resulting morphs are also closer to the worst case when evaluating with CurricularFace and vice versa. When either of the two is used for optimisation, they are no less vulnerable to GAN-based morphing attacks than other FR systems. Interestingly, Dlib—and to a lesser extent also the COTS FR system—is less vulnerable to MIPGAN and WALI morphs than to landmark morphs. It is also interesting to highlight the inverse relationship between performance on normal images and vulnerability to morphing attacks. Comparing the last two columns illustrates this in theory: the two FR systems with the lowest FNMR also have the highest worst-

Table 1: MMPMR for WALI morphs without any optimisation steps.

	MIPGAN	WALI (Ours) with all FR losses	WALI without FR losses	WALI w/o $\mathscr{L}_{\text{FR\_Morph\_}\alpha}$	WALI w/o $\mathscr{L}_{\text{FR\_Morph}}$	WALI w/o $\mathscr{L}_{\text{FR}}$
MobileFaceNet	0.9	19.2	0.3	**19.6**	19.3	14.4
ElasticFace (black box)	0.0	0.0	0.0	0.0	0.0	0.0
Curricularface (black box)	0.0	0.0	0.0	0.0	0.0	0.0
ArcFace (black box)	0.7	**13.0**	0.1	12.0	6.9	6.6
Inception (black box)	0.5	**13.2**	0.6	7.9	9.5	9.0
PocketNet (black box)	1.5	**18.0**	1.1	16.8	17.7	14.0
VGG16 (black box)	2.0	**10.7**	0.4	9.5	8.0	6.0
Dlib (black box)	5.6	**10.1**	0.3	8.1	3.7	7.5
COTS (black box)	0.0	0.1	0.0	**1.5**	**1.5**	1.4

Note. The more challenging the morphs, the higher the MMPMR. The highest value in each row is shown in bold.

Table 2: MMPMR values for landmark- and GAN-based morphs.

	Land-mark	MIPGAN	Improved MIPGAN (Ours)	WALI (Ours) with optimisation using							Worst Case	FNMR
				MFN	*MFN & El.Face*	*MFN & Curr.Face*	*MFN & ArcFace*	*MFN & Inception*	*MFN & PocketNet*	*MFN & VGG16*		
MobileFaceNet	65.7	71.9	**91.8**	96.8	**96.6**	**96.6**	96.8	97.0	97.0	96.6	*97.5*	*0.5*
ElasticFace	56.9	18.9	**83.0**	14.0	**81.8**	**60.4**	25.7	20.8	18.7	18.1	*98.8*	*0.0*
CurricularFace	45.9	11.1	**60.9**	8.6	**46.6**	**68.3**	14.8	12.2	10.5	13.2	*99.0*	*0.0*
ArcFace	70.7	62.9	**84.5**	64.6	**76.2**	**75.6**	90.4	71.7	70.2	70.2	*97.9*	*0.2*
Inception	36.8	37.0	**51.1**	37.6	**47.2**	**46.4**	43.7	58.0	41.2	42.5	*71.8*	*3.4*
PocketNet	34.1	34.2	**49.0**	48.0	**49.3**	**48.7**	51.4	51.3	63.5	50.3	*84.2*	*3.8*
VGG16	36.4	32.7	**42.1**	35.4	**39.4**	**40.1**	40.1	41.1	38.5	56.2	*92.0*	*7.6*
Dlib	45.1	37.2	**42.4**	27.3	**32.6**	**31.4**	32.5	32.9	30.0	32.2	*72.3*	*5.8*
COTS	99.8	93.4	**98.6**	71.4	**94.6**	**95.5**	79.6	80.4	76.3	75.0	*n/a*	*0.0*

Note. The second-to-last column shows the theoretical worst case for each respective FR system. Underlined numbers indicate evaluation was under white-box assumptions, i.e., this FR system was used during optimisation. The more challenging the morphs, the higher the MMPMR. To show that there is a trade-off between FR performance and vulnerability to morphing attacks, we report the false non-match rate (FNMR) (%) at which the false match rate <0.1% in the last column. The morphing methods highlighted in bold are closest to the worst case for almost all FR systems.

case MMPMR. In practice, the same pattern is shown: the FR systems with lower FNMR are indeed more vulnerable to landmark, MIPGAN, and WALI morphs.

Table 1 reports the MMPMR and confirms our hypothesis that explicitly considering the goal of morphing *during* training leads to more challenging morphs. There may be some amount of trade-off between the two goals when using WALI: generating visually convincing images versus successfully manipulating identity information.

The following four aspects lead to more challenging morphs:

(1) defining a worst-case embedding that we can use to define losses during training and optimisation,
(2) explicitly training the model to generate morphs,
(3) improving optimisation by splitting it into two phases: before we generate morphs, we select good initial embeddings for each input image,
(4) optimising with more than one FR system.

WALI does not seem to generalise well to other datasets. This can be seen in Table 3 and Figure 6. This is to a large extent due to our WALI Generator (7.8 million parameters) not being able to compete with a more powerful generator such as StyleGAN (28.3 million parameters). When applying a colour correction to FRLL images so that they more closely resemble FRGC images, the MMPMR of WALI morphs significantly increases, for example, from 30.0% to 11.0% for CurricularFace or 14.1%–44.5% for PocketNet, indicating that the lower performance of WALI is to a large extent due to the different type of data. In order to illustrate the effect of approximating a worst-case when considering FRLL data, we can apply three of the four improvements listed above to existing generative methods. We show that combining a more powerful StyleGAN Generator with the improved optimisation approach in two phases, as well as optimising with two FR systems, still leads to closer approximations of worst-case morphs. Morphs generated with our improved MIPGAN implementation have higher MMPMR values than all other GAN-based morphs and also higher MMPMR than AMSL morphs. While the MMPMR for the other three landmark-based methods is higher, those morphs contain very obvious artefacts. Since the MIPGAN optimisation process includes a perception-style loss that encourages visual similarity to both contributing identities, the MIPGAN morphs contain some ghosting artefacts. Because we do not include such a loss during optimisation Phase 2, the improved MIPGAN morphs are visually more convincing than MIPGAN morphs and

Table 3: MMPMR for FRLL morphs.

	Landmark-based morphing				GAN-based morphing					
	AMSL	Face-Morpher	OpenCV	WebMorph	Style-GAN	MIPGAN	Improved MIPGAN (Ours)	WALI (Ours) MFN&EF w/o colourcorr.	WALI (Ours) MFN&EF w colourcorr.	Worst-case
MobileFaceNet	64.7	89.2	86.0	90.3	22.1	74.4	**96.2**	96.7	96.8	*99.5*
ElasticFace	38.8	58.8	60.4	60.0	0.0	4.7	**74.4**	26.1	44.0	*99.8*
Curricularface	32.7	53.4	55.6	54.3	0.0	2.6	**44.3**	3.0	11.0	*99.6*
ArcFace	58.8	67.2	64.4	65.3	1.2	20.3	**64.6**	7.8	17.0	*99.4*
Inception	9.6	14.2	14.2	17.3	0.3	5.0	**11.2**	1.2	2.4	*49.3*
PocketNet	51.7	78.0	80.3	84.7	20.0	55.9	**75.8**	14.1	44.5	*98.8*
VGG16	12.1	12.8	14.2	25.2	0.4	6.8	**20.5**	2.3	4.7	*100.0*
Dlib	0.9	1.0	1.0	0.9	0.0	0.0	**0.6**	0.1	0.1	*16.5*
COTS	97.6	100.0	100.0	100.0	0.8	70.4	**96.6**	46.6	64.7	*n/a*

Note. The more challenging the morphs, the higher the MMPMR. Underlined numbers indicate the FR system was used for optimisation. Of all GAN-based approaches the improved MIPGAN approach (highlighted in bold) is closest to the worst case for almost all FR systems.

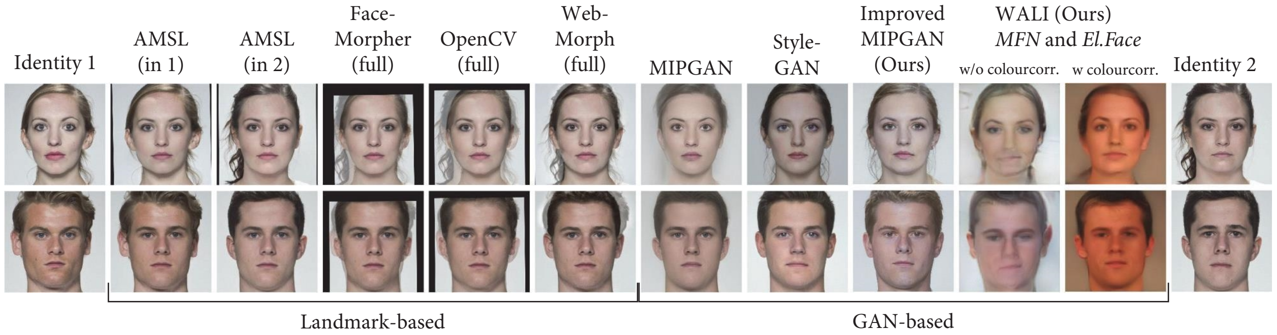


Figure 6: Examples of morphs based on FRLL images. WALI (and other morph methods) are trained on another dataset and applied to FRLL images, which have different lighting and colour balance. WALI may not generalise well to unseen data, mainly because of the simple WALI generator, which cannot compete with more powerful GANs. Incorporating StyleGAN in our WALI pipeline results in "Improved MIPGAN," giving visually convincing results.

landmark morphs that contain visible ghosting artefacts. Some of the other three landmark-based approaches outperform improved MIPGAN but contain significant ghosting artefacts that would not fool visual inspection by humans. If a large network such as StyleGAN were explicitly *trained* to generate morphs, they might become even more challenging.

6.1. S-MAD Using LBP. We implement an S-MAD approach based on LBP followed by an SVM. We compare the ability of this model to detect morphs: once when it was trained using the same training data as WALI and once when using a separate training set. LBP features may be appropriate for detecting WALI-based morphs when the underlying training data are known, but performance decreases significantly when the database is unknown and contains only landmark-based morphs, see Table 4 and Figure 7. LBP features are not at all suitable for detecting (improved) MIPGAN morphs, which is probably due to the ability of StyleGAN to generate images with texture that is similar to that of real images. The APCER for images generated by a 512 × 512 WALI model trained without FR losses, see the bottom row in Figure 4 and the last column in Table 4, ranges from 78.8% to 99.8%, showing a similar effect.

6.2. D-MAD Using Deep-Learning-Based FR Feature Differences. While this approach seems to be very successful at detecting morphed images that were created using the same algorithm that was used to create the training set, its performance decreases significantly when evaluating (improved) MIPGAN or WALI morphs, see Table 4 and Figure 7. Note that this D-MAD approach can detect images generated by a 512 × 512 WALI model trained without FR losses much more easily than other GAN-based morphs, which makes sense, since these morphs were not optimised using FR systems. If this approach were trained with a separate training set other than FRGC, we would expect its performance on MIPGAN or WALI morphs to decrease further.

6.3. Discussion. For all FR systems we evaluated, except Dlib, our approach outperforms MIPGAN morphs based on FRGC. For three out of six FR systems tested under black-box assumptions, WALI morph outperform landmark morphs. This shows that it is possible to approximate the theoretical worst case for more than one FR system. As we already mentioned, this does not mean that ElasticFace and CurricularFace are generally less vulnerable to GAN-based morphing attacks. These two FR

Table 4: Detection performance in BPCER (%) at APCER ≤5% and ≤10% (Section 5.1).

Trained with	Land-mark	MIPGAN	Improved MIPGAN (Ours)	WALI (Ours) with optimisation using:							WALI (Ours) 512 × 512 baseline
				MFN	*MFN & El.Face*	*MFN & Curr.Face*	*MFN & ArcFace*	*MFN & Inception*	*MFN & PocketNet*	*MFN & VGG16*	
S-MAD	BPCER@APCER ≤5%										
FRGC	3.2	100.0	99.4	71.5	66.2	67.0	69.2	70.9	71.2	63.9	81.3
AMSL	38.7	100.0	95.2	54.2	62.4	48.8	45.9	48.4	48.0	42.6	99.8
	BPCER@APCER ≤10%										
FRGC	1.4	100.0	98.6	56.0	52.2	50.7	55.9	49.8	45.5	47.9	78.8
AMSL	26.2	100.0	89.0	29.5	38.1	31.4	26.5	26.9	26.7	26.9	99.4
D-MAD	BPCER@APCER 5%										
FRGC	0.3	19.4	18.5	7.2	9.8	10.2	12.8	8.3	6.7	8.4	0.5
	BPCER@APCER ≤10%										
FRGC	0.2	12.4	12.0	3.5	5.1	5.6	7.3	4.8	3.7	4.2	0.2

Note. Top = LBP-based S-MAD. Bottom = D-MAD based on FR-difference features.

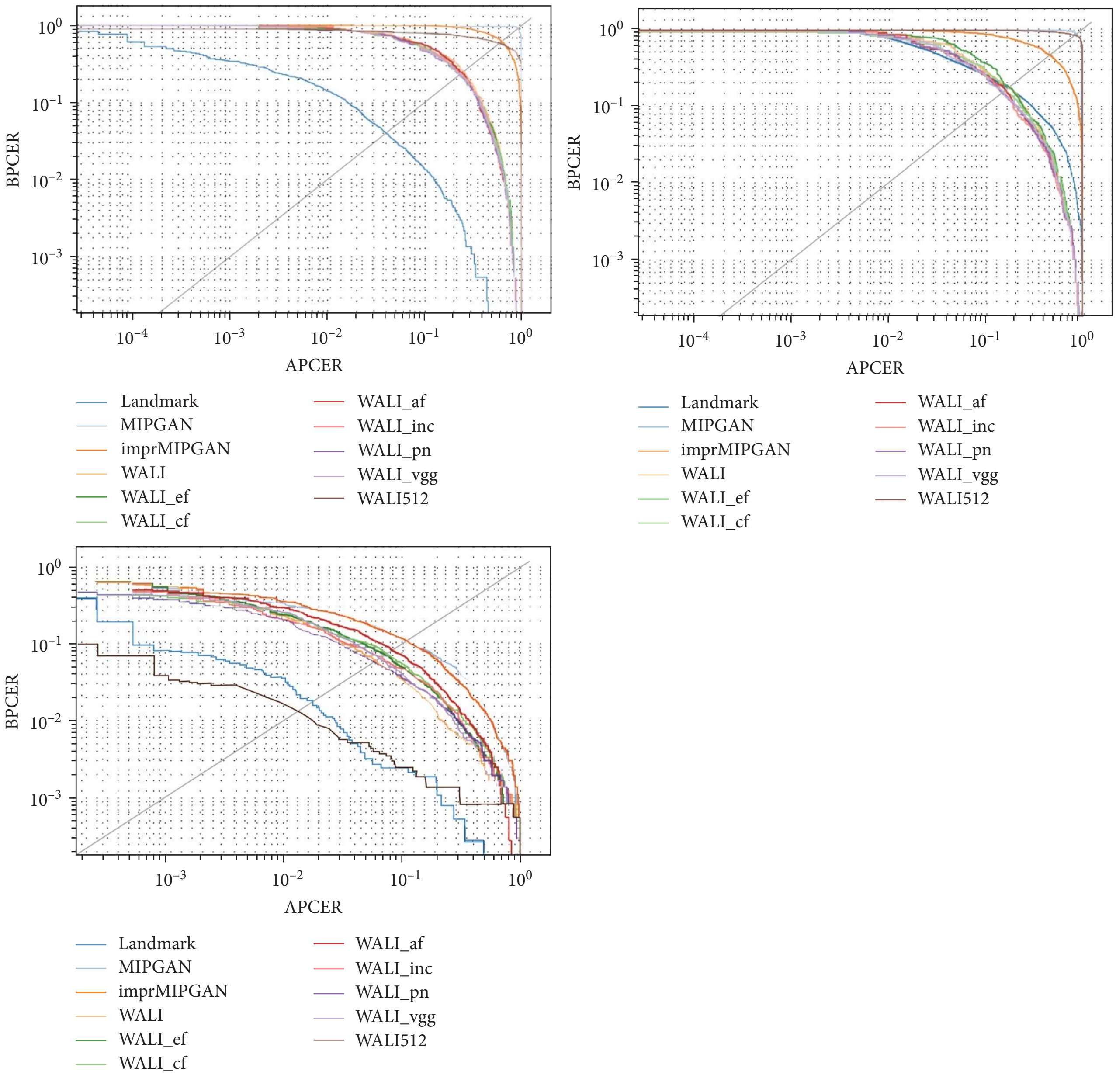


Figure 7: DET curves. Top: LBP-based SMAD trained with FRGC data. Middle = LBP-based SMAD trained with FRLL data. Bottom = DMAD based on FR feature difference trained with FRGC data.

systems are newer and seem to show different behaviour from the other FR systems we tested.

Using WALI to generate morphs is computationally expensive, since optimisation needs to be performed for every morph that is generated. Due to hardware limitations, we report results for 128 × 128 images. While we did successfully generate larger images—up to 512 × 512, compared to MIPGAN morphs that rely on a StyleGAN model that generates 1024 × 1024 images—this takes significantly longer and requires more GPU memory, especially during training. However, our results do show that morphs exist that are extremely successful at exploiting the vulnerabilities of (multiple) FR systems. Therefore, the idea of a criminal tweaking their morph in ways to make it more likely to be accepted by multiple FR systems is very possible, illustrating the need to focus on quality as well as quantity when generating morphing datasets. We evaluated five different FR systems and showed that in the (theoretical) worst case up to 72%–98% of FRGC morphs can trick the FR system. For four out of five FR systems we evaluated, our WALI morphs, when optimised with two FR systems, are closer to this upper bound than either landmark or MIPGAN morphs. As has been reported before [11], there seems to be an inverse relationship between the performance of FR systems on normal data and vulnerability to morphing attacks.

WALI's improvements are due to having a worst-case embedding as a goal to approximate improved optimisation in two phases (finding a good initial embedding for each bona fide image before generating morphs), optimising with more than one FR system simultaneously and including the goal of morphing during training. The first three goals can be applied to other existing generative methods; we used StyleGAN as an example, leading to an improved MIPGAN approach that led to morphs that are more challenging than other GAN-based morphs.

WALI morphs were generated in an adversarial manner and probably exploited the fact that deep-learning-based FR systems are sensitive to certain patterns in images. While such patterns might be imperceptible to humans, they can make the FR systems vulnerable to WALI morphs. These patterns may not survive post-processing, such as printing and scanning, resizing, etc. Furthermore, there are still artefacts visible to the human eye, as can be seen in Figures 4 and 5, for example, around the mouth or eyes. Visual inspection would probably allow, e.g., border guards to detect that the generated morph is not a real image. Our findings, therefore, show room for improvement for FR systems. We hope that our proposed method WALI can contribute to such an improvement by generating more challenging training data for FR systems.

7. Conclusion and Future Work

In this work, we showed that generating challenging morphs is possible and necessary to evaluate the robustness of FR systems. Our newly proposed WALI method outperformed existing morphing techniques on FRGC data, and since it provides a way to generate large quantities of difficult morphs, it could contribute to improving FR and MAD systems' performance. We also introduced an improved MIPGAN approach that, due to the powerful underlying StyleGAN Generator, generated challenging morphs on FRLL as well as on FRGC. We showed that if the goal of generating challenging morphs is not explicitly considered during the training of a GAN, then the resulting morphs will be significantly less challenging than when that goal is included during training.

Challenges for future research include generating such datasets while also making sure to cover the possible range of morphs by focussing on (visual) quality as well as quantity, for example, by investigating the effect of time-consuming manual postprocessing. It would be interesting to explore whether GAN networks that can produce images with as high quality as, e.g., StyleGAN can also be adapted to explicitly include the goal of generating difficult morphs during training. We showed that optimising towards a worst-case leads to more challenging morphs; similar adaptations could be made to diffusion-based approaches as well. Additionally, further investigation could be carried out on the effect of post-processing techniques on the robustness of FR systems to morphs. Moreover, the effects of training FR systems or MAD methods with large datasets generated with WALI or improved MIPGAN could be further explored in future research.

8. Ethics, Broader Impact, and Reproducibility

This paper introduces methods to generate morphs, which could potentially be used to apply for passports or other documents that could be shared by two people, for example, allowing them to avoid travel restrictions. As long as countries allow applicants to provide their own digital or printed passport photo, this will continue to pose a risk. On the other hand, sharing our morphing generation method will allow researchers to be more aware of potential vulnerabilities and support the development of countermeasures. Our method can be used to generate large datasets of advanced morphs that can, for example, be used to train FR systems or to teach human border control staff to better spot morph-related artefacts. We aim to raise awareness for risks posed by morphing, and without sharing our method, such vulnerabilities might remain unknown. We also intend to share our code for research purposes only. To aid reproducibility, we have included important information, such as hyperparameters, in this paper. All data we used is already available to researchers, and we plan to release our code for research purposes after publication.

Data Availability

The data used in this study are available at FRGC: https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc FFHQ: https://github.com/NVlabs/ffhq-dataset FRLL: https://omen.cs.uni-magdeburg.de/disclaimer/index.php; https://www.idiap.ch/en/dataset/frll-morphs.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Rijksdienst voor Identiteitsgegevens (RvIG). We thank Prof. Dr. Christoph Brune and Dr. Jelmer Wolterink for useful discussions.

References

[1] U. Scherhag, A. Nautsch, C. Rathgeb et al., "Biometric systems under morphing attacks: assessment of morphing techniques and vulnerability reporting," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7, 2017.

[2] D. J. Robertson, R. S. S. Kramer, and A. M. Burton, "Fraudulent ID using face morphs: experiments on human and automatic recognition," *PLoS One*, vol. 12, no. 3, Article ID e0173319, 2017.

[3] April 2023, https://www.computerbild.de/artikel/cb-News-Panorama-Neues-Gesetz-Passbilder-kuenftig-nur-noch-digital-26305181.html.

[4] April 2023, https://www.dfa.ie/passportonline/onlinephotoguidelines/.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, pp. 1–9, Curran Associates, Inc, 2014.

[6] N. Damer, A. M. Saladié, A. Braun, and A. Kuijper, "Morgan: recognition vulnerability and attack detectability of face morphing attacks created by generative adversarial network," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–10, 2018.

[7] H. Zhang, S. Venkatesh, R. Ramachandra, K. Raja, N. Damer, and C. Busch, "MIPGAN—generating strong and high quality morphing attacks using identity prior driven gan," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 365–383, 2021.

[8] N. Damer, M. Fang, P. Siebke, J. N. Kolf, M. Huber, and F. Boutros, "Mordiff: recognition vulnerability and attack detectability of face morphing attacks created by diffusion autoencoders," in *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, 2023.

[9] Z. Blasingame and C. Liu, "Leveraging diffusion for strong and high quality face morphing attacks," 2023.

[10] K. Raja, G. Gupta, S. Venkatesh, R. Ramachandra, and C. Busch, "Towards generalized morphing attack detection by learning residuals," *Image and Vision Computing*, vol. 126, 2022, https://www.sciencedirect.com/science/article/pii/S0262885622001640, Article ID 104535.

[11] L. Colbois and S. Marcel, "On the detection of morphing attacks generated by gans," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, 2022.

[12] U. M. Kelly, L. Spreeuwers, and R. Veldhuis, "Worst-case morphs: a theoretical and a practical approach," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, 2022.

[13] K. Raja, M. Ferrara, A. Franco et al., "Morphing attack detection – database, evaluation platform and benchmarking," *Computer Vision and Pattern Recognition*, 2020, https://arxiv.org/abs/2006.06458v3.

[14] NIST, "NIST FRVT MORPH," October 2021, https://pages.nist.gov/frvt/html/frvt_morph.html.

[15] BOEP, "Bologna online evaluation platform (BOEP)-morph attack detection evaluation," October 2021, https://biolab.csr.unibo.it/fvcongoing/UI/Form/BOEP.aspx.

[16] V. Dumoulin, I. Belghazi, B. Poole et al., "Adversarially learned inference," 2017.

[17] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019.

[18] N. Damer, K. Raja, M. Süßmilch et al., "Regenmorph: visibly realistic gan generated face morphing attacks by attack re-generation," 2021, https://arxiv.org/abs/2108.09130.

[19] F. Vakhshiteh, A. Nickabadi, and R. Ramachandra, "Adversarial attacks against face recognition: a comprehensive study," *IEEE Access*, vol. 9, pp. 92735–92756, 2021.

[20] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "GAN inversion: a survey," 2021.

[21] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2017.

[22] C. Seibold, A. Hilsmann, and P. Eisert, "Style your face morph and improve your face morphing attack detector," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–6, 2019.

[23] M. Ferrara, A. Franco, and D. Maltoni, "Face morphing detection in the presence of printing/scanning and heterogeneous image sources," *IET Biometrics*, vol. 10, no. 3, pp. 290–303, 2021, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/bme2.12021.

[24] S. Venkatesh, K. Raja, R. Ramachandra, and C. Busch, "On the influence of ageing on face morph attacks: Vulnerability and detection," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–10, 2020.

[25] U. Scherhag, J. Kunze, C. Rathgeb, and C. Busch, "Face morph detection for unknown morphing algorithms and image sources: a multi-scale block local binary pattern fusion approach," *IET Biometrics*, vol. 9, no. 6, pp. 278–289, 2020, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-bmt.2019.0206.

[26] S. Akash, V. Lazar, R. Chris, U. G. M., and S. Charles, "VEEGAN: reducing mode collapse in gans using implicit variational learning," *Neural Information Processing Systems*, 2017.

[27] C. Li, H. Liu, C. Chen et al., "ALICE: towards understanding adversarial learning for joint distribution matching," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, and S. Bengio, et al., Eds., vol. 30, Curran Associates, Inc, 2017.

[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, http://arxiv.org/abs/1701.07875.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, and S. Bengio, et al., Eds., vol. 30 of *NIPS'17*, pp. 1–11, Curran Associates, Inc, 2017.

[30] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016, http://distill.pub/2016/deconv-checkerboard.

[31] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Focal frequency loss for image reconstruction and synthesis," in *ICCV*, 2021.

[32] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: efficient cnns for accurate real-time face verification on mobile devices," *CoRR*, vol. 1804, no. 7573, 2018, http://arxiv.org/abs/1804.07573.

[33] "Vggface weights," 2018, https://github.com/rcmalli/keras-vggface.

[34] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, 2019.

[35] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," *CoRR*, vol. 1503, no. 3832, 2015, http://arxiv.org/abs/1503.03832.

[36] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Elasticface: elastic margin loss for deep face recognition," *CoRR*, vol. 2109, no. 9416, 2021, https://arxiv.org/abs/2109.09416.

[37] Y. Huang, Y. Wang, Y. Tai et al., "Curricularface: adaptive curriculum learning loss for deep face recognition," *CoRR*, vol. 2004, no. 00288, 2020, https://arxiv.org/abs/2004.00288.

[38] F. Boutros, P. Siebke, M. Klemt, N. Damer, F. Kirchbuchner, and A. Kuijper, "Pocketnet: extreme lightweight face recognition network using neural architecture search and multi-step knowledge distillation," *CoRR*, vol. 2108, no. 10710, 2021, https://arxiv.org/abs/2108.10710.

[39] D. E. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[40] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, Y. Bengio and Y. LeCun, Eds., Conference Track Proceedings, San Diego, CA, USA, http://arxiv.org/abs/1412.6980, May 2015.

[41] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *Computer Vision and Pattern Recognition*, 2021.

[42] A. Paszke, S. Gross, F. Massa et al., "Pytorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, Curran Associates, Inc, 2019.

[43] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: efficient CNNs for accurate real-time face verification on mobile devices," *CoRR*, vol. 1804, no. 7573, 2018, http://arxiv.org/abs/1804.07573.

[44] P. J. Phillips, P. J. Flynn, T. Scruggs et al., "Overview of the face recognition grand challenge," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 947–954, 2005.

[45] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel, "Are gan-based morphs threatening face recognition?" in *ICASSP. 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2959–2963, 2022.

[46] P. Korshunov, L. Colbois, and S. Marcel, "Vulnerability analysis of face morphing attacks from landmarks and generative adversarial networks," October 2020, https://arxiv.org/abs/2012.05344.

[47] L. DeBruine and B. Jones, "Face Research Lab London set," May 2017, https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666/3.

[48] M. Ferrara, A. Franco, D. Maltoni, and C. Busch, "Morphing attack potential," in *2022 International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, 2022.

[49] R. Raghavendra, K. B. Raja, and C. Busch, "Detecting morphed face images," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–7, 2016.

[50] L. Spreeuwers, M. Schils, and R. Veldhuis, "Towards robust evaluation of face morphing detection," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1027–1031, 2018.

[51] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch, "Deep face representations for differential morphing attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3625–3639, 2020.