MorphGANFormer: Transformer-based Face Morphing and De-Morphing

Na Zhang, Xudong Liu, Xin Li, *Fellow, IEEE*, Guo-Jun Qi

***Abstract*—Semantic face image manipulation has received increasing attention in recent years. StyleGAN-based approaches to face morphing are among the leading techniques; however, they often suffer from noticeable blurring and artifacts as a result of the uniform attention in the latent feature space. In this paper, we propose to develop a transformer-based alternative to face morphing and demonstrate its superiority to StyleGAN-based methods. Our contributions are threefold. First, inspired by GANformer, we introduce a bipartite structure to exploit long-range interactions in face images for iterative propagation of information from latent variables to salient facial features. Special loss functions are designed to support the optimization of face morphing. Second, we extend the study of transformer-based face morphing to demorphing by presenting an effective defense strategy with access to a reference image using the same generator of MorphGANFormer. Such demorphing is conceptually similar to unmixing of hyperspectral images but operates in the latent (instead of pixel) space. Third, for the first time, we address a fundamental issue of vulnerability-detectability trade-off for face morphing studies. It is argued that neither doppelganger nor random pair selection is optimal, and a Lagrangian multiplier-based approach should be used to achieve an improved trade-off between recognition vulnerability and attack detectability.**

***Index Terms*—transformer, face morphing, De-morphing.**

I. Introduction

With the rapid development of deep-learning technology, automatic face recognition (FR) has become a key method in security-sensitive applications of identity management (e.g. travel documents). However, the face recognition system (FRS) is vulnerable to face morphing attacks [1], which aim to create facial images that can be successfully matched to more than one person. Existing face-morphing methods can be classified into two categories. One is performed on the image level via landmark interpolation, like OpenCV [2], FaceMorpher [3], LMA [4], WebMorph [5]. The other works are performed by manipulating latent codes of generative adversarial networks (GAN), such as MIPGAN-II [6], MorGAN [4], StyleGAN [7]. Both approaches have serious limitations. For landmark-based methods, as the morphing process translates landmarks and the associated texture, misaligned pixels tend to generate artifacts and ghost-like images, making the images unrealistic (i.e., easy for a human observer to detect). Similarly, for GAN-based methods, unpleasant visual artifacts, such as noticeable blurring and abnormal image patterns, often occur, often making morphed faces unnatural (see Fig. 1). It is natural to seek an alternative approach to face morphing attacks.

Transformer-based architectures have found successful applications in natural language processing [8]–[10], object detection [11], image restoration [12], [13], video inpainting [14], [15], image synthesis [16]–[21], and so on. Inspired by the capability of exploiting the long-range dependency of GANformer [17], we propose to develop the GANformer-based morphing attack in a compositional latent space, as shown in Fig. 1 (b). The compositional latent space is composed of multiple latent components in local-style and one latent component in global-style, respectively. Such a compositional design allows us to have finer control of salient regions (e.g., face in the foreground) than the less important region (e.g., background). Meanwhile, MorphGANFormer is bidirectional, allowing the propagation of information between latent codes and image features in both directions. In addition to long-range dependency, duplex attention on bipartite graphs facilitates the synthesis of high-resolution by keeping computation linear.

Under the transformer-based framework, we focus on the design of latent code in the compositional space. Unlike GANformer [17] which simply adopts the loss function of StyleGAN studies [7], [22], we have designed a class of loss functions specifically tailored for face morphing applications. Our design attempts to expedite the search for a suitable latent code by combining the strengths of both landmark-based and GAN-based approaches. Both facial landmarks and features (e.g., histogram of orientated gradients [23]) are included as content-related regularization terms. Style-related regularization consists of VGG-based perceptual loss and pixel-based MSE loss. The tradeoff between the style and context loss terms allows us to strike an improved balance between visual quality (i.e., fewer artifacts) and attack success (i.e., better matching).

Like other security systems, morphing attacks and defenses co-evolve in a never-ending race. Morphing and demorphing [24], [25] are two sides of the same coin, although relatively less attention has been paid to demorphing studies in the literature. The other contribution of this work is to conduct a dual study of demorphing in latent space, which complements our construction of MorphGANFormer. For the first time, we address a fundamental issue of vulnerability-detectability tradeoff for face morphing studies - i.e., what pair of images should be used in morphing study? A pair of similar images (e.g., doppelganger [26]) might be desirable from a recognition vulnerability perspective but suffers from being more easily detectable (i.e., higher APCER/BPCER rates). On the other hand, two random faces enjoy the advantage from the attack detectability perspective, but sacrifice the recognition vulnerability (i.e., lower MMPMR rate [27]). It is argued that neither the selection of doppelgangers nor random pairs is optimal and a Lagrangian multiplier-based approach should be

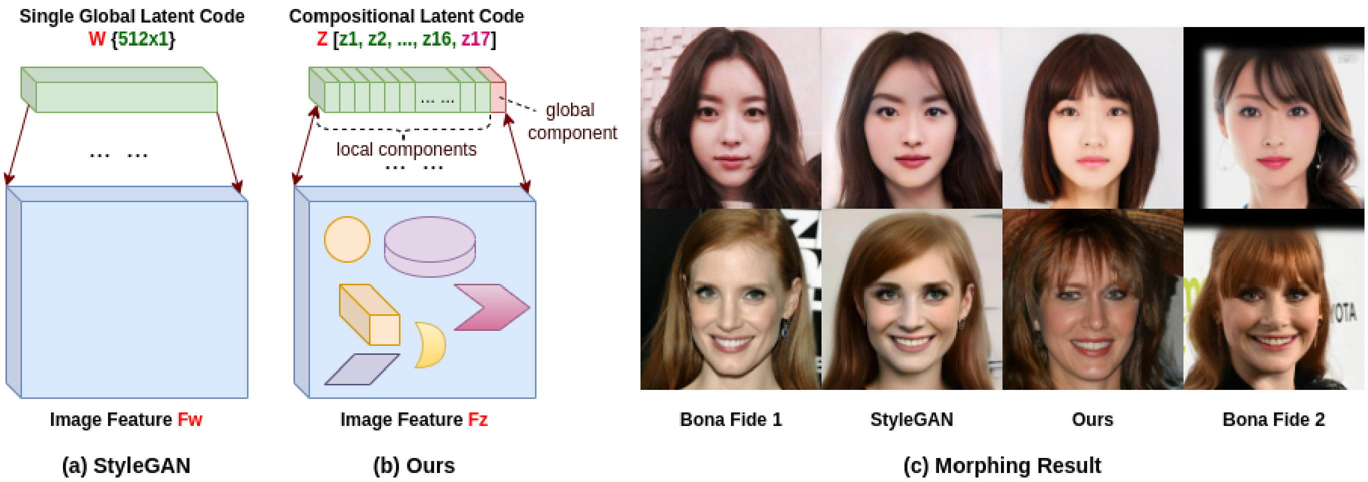


Fig. 1. Illustration of latent code modulation of (a) StyleGAN and (b) Our MorphGANFormer. StyleGAN uses a single global-style latent code to modulate the whole scene uniformly in one direction. Ours is a compositional latent code with 16 local- and one-global-style components to impact different regions in the image allowing for spatially finer control over the generation process bidirectionally. Figure (c) shows some morphing results of StyleGAN-based model and our MorphGANFormer (ours contain fewer visual artifacts).

used to achieve an improved trade-off between the recognition vulnerability and the detectability of the attack [4]. The main contributions of this paper are summarized below.

- Design a transformer-based GAN model with a compositional latent space. It is made up of 16 local-style latent code components and one extra global-style component with 32×1 dimension for each. Different components can impact different regions in the image, allowing for spatially finer control over the generation process bidirectionally.
- Design special loss functions to improve the performance of the latent code optimization problem by maximizing the similarity between the generated face and the target face. Four types of loss function are adopted: biometric loss, landmark-based loss, perceptual loss, and pixel-wise mean square error (MSE).
- Extend the study of transformer-based face morphing to demorphing using the same generator. With the final morphed face and a given trusted live capture of one bona fide face, we have shown how to successfully restore the other bona fide face.
- Experimental results with both Doppelganger and random selection to demonstrate the trade-off between recognition vulnerability and attack detectability. We hope that this line of research will lead to a deeper understanding of adversarial attack and defense in the study of face morphing and demorphing.

II. Related Works

A. Landmark-based Generation

Morphed face is initially performed by detecting facial landmarks of two bona fide faces. The final morphed face is generated by landmark interpolation and texture blending. The landmark-based method, as the name suggests, works by obtaining landmark points on facial regions, like the nose, eyes, mouth, etc. The landmarks obtained from two bona fide faces are warped by moving the pixels to different, more averaged positions. There exist different procedures for warping in the literature. Delaunay triangulation is a popular one. The basic idea is to perform Delaunay Triangulation on the three sets of landmarks (2 bona fides and their average points) and do affine transform and warping. The two warped faces will do alpha blending, and then the final morphed face is generated.

The most popular methods contain OpenCV [2], FaceMorpher [3], LMA [4], WebMorph [5], etc. In the OpenCV [2] algorithm, the landmarks of the bona fide faces are obtained by Dlib [28] and then used to form Delaunay triangles [29], which in turn are warped and mixed with alpha. FaceMorpher [3] is also an open-source tool similar to OpenCV, but with the STASM [30] landmark detector instead. Both algorithms create morphs with noticeable ghosting artifacts, as the region outside the area covered by these landmarks is simply averaged. WebMorph [5] is an online landmark-based morphing tool that requires 189 landmarks, to generate morphed images with better alignment and of higher visual quality. Ghosting artifacts are still visible and prominent around the hair and neck area. Similar to OpenCV and FaceMorpher, LMA [4] is performed by detecting facial landmarks, the mean face points for each image are calculated and each image is then warped to sit on these coordinates after performing the Delaunay triangulation, but uses 194 points detected by an ensemble of randomized regression trees [31]. One special is a combined private Morphs tool used in the AMSL face morph image database [32]. This tool can generate very realistic morphs with virtually no ghosting artifacts, even around the hair and neck area, thanks to the additional Poisson image editing.

B. GAN-based Generation

GAN-based model has made a major breakthrough in high-quality image synthesis, especially on human faces [22]. Taking advantage of the advanced GAN architectures and their ability to produce synthetic images, we proposed a few GAN-based morphing approaches that avoid image-level interpolation. It works by embedding the images into the

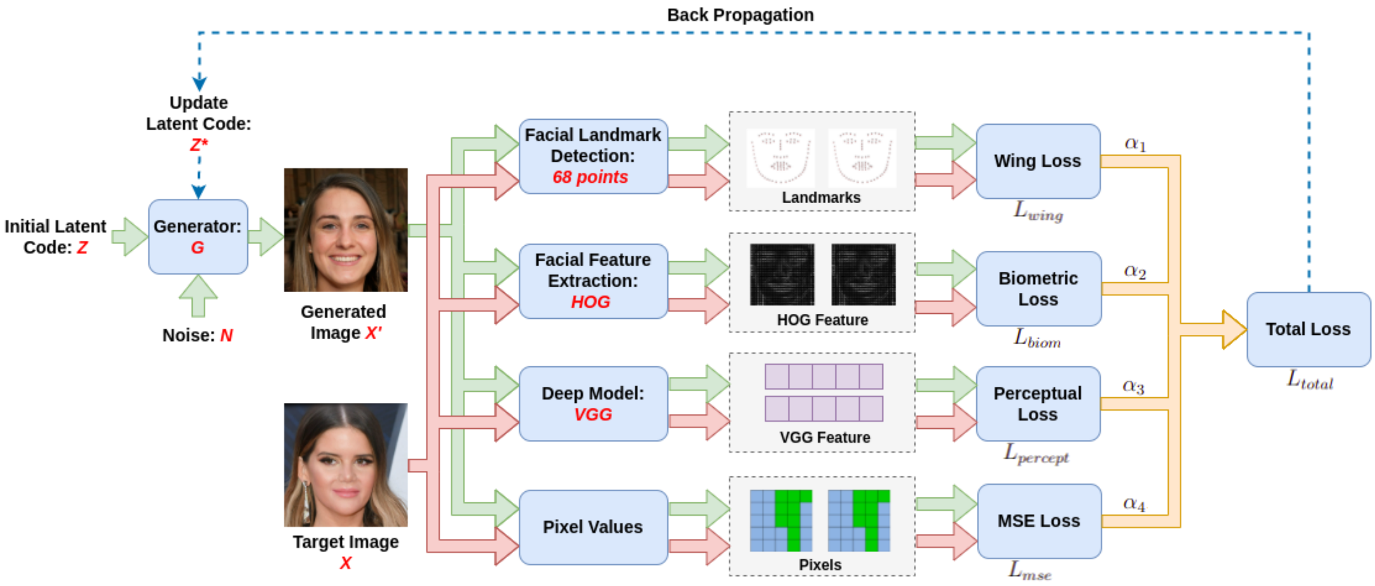


Fig. 2. The pipeline of optimizing the latent code of the given face image.

intermediate latent space. First, two bona fide face images are mapped into the latent space to obtain their latent codes. And then a linear combination of two latent codes is made to obtain a final latent code, which is put into the generator of the pre-trained GAN model to synthesize the morphed image.

StyleGAN2 [7] is a morphing algorithm that can generate realistic high-resolution faces. Based on StyleGAN [22], the MIPGAN-II [6] was designed to generate images with higher identity preservation by introducing a loss to optimize identity preservation in the latent vector. MorGAN [4] is based on automatic image generation using a specially designed GAN. An enhanced version called CIEMorGAN [33] has also been released.

C. De-Morphing

The common definition of demorphing is that by using one bona fide identity as a reference image, the morphed face image can be reverted (or demorphed) to reveal the identity of the other bona fide subject. In [24], the authors reverse the morphing operation to find the second bona fide by exploiting the live image acquired from the first bona fide. In FD-GAN [34], the authors designed a symmetric dual network and adopted two layers of restoration losses to separate the second bona fide's face image. The basic idea is that it first restores the image of the second bona fide from the given morphed input using the first bona fide as a reference, and then tries to restore the first bona fide from the morphed image with the restored second bona fide as a reference. In [35], a conditional GAN is designed to disentangle identity from the morphed image using the pixel difference by minimizing conditional entropy. Recently, [36] proposed a method to recover both bona fide face images simultaneously from a single given morphed face image without reference image or prior knowledge. Such blind demorphing is conceptually similar to the unmixing of hyperspectral images.

In addition, some works have been proposed that treat face demorphing as a technique to detect reference-based morphing attacks [37], [38]. For example, in [38], the authors apply a fusion of two differential morphing attack detection methods, i.e., demorphing and deep-face representations, for detection. [25] focuses on the robustness of face demorphing and uses it as a technique to protect face recognition systems against the well-known threat of morphing.

III. Methodology

A. Transformer-based GAN

Most existing GAN-based models adopt CNN as the basic architecture and rarely consider self-attention constructions. In this work, we have designed a transformer-based GAN model aiming to eliminate the blending artifacts, as well as, eliminate the manipulation in the latent space, resulting in more visibly realistic morphed faces. We applied the Generative Adversarial Transformer (GANformer) [17] as our backbone to generate high-quality morphing face images with 1024×1024 resolution by linearly interpolating the latent codes of the two input bona fide faces. The latent code is generated by improving the similarity between the input bona fide image and the embedded image created using a latent vector. In our work, we call the MorphGANFormer morphing model.

MorphGANFormer contains a generator (G) that maps a sample from the latent space to an image, and a discriminator (D) that seeks to discern between real and fake images [39]. G and D compete with each other through a minimax game until they reach equilibrium [17]. The generator employs a bipartite structure, called bipartite transformer. Traditional transformer uses self-attention with pairwise connectivity, as shown in Fig. 3 (a). It is a highly-adaptive architecture centered around relational attention and dynamic interaction. However, the dense and potentially excessive pairwise connectivity causes quadratic mode of operation making it difficult to be extended to high-resolution input image. Bipartite transformer adopts a point-to-point mapping between individual latent components and different regions of evolving visual features, which can

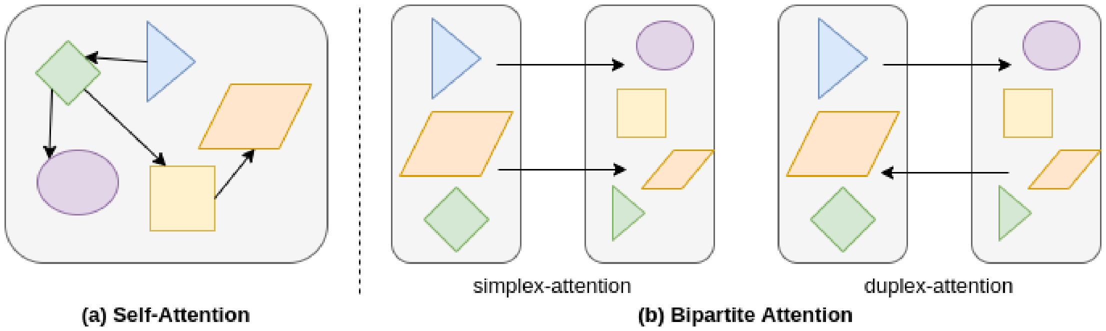


Fig. 3. Self-Attention (a) and Bipartite Attention (b). In comparison to self-attention, bipartite attention allows long-range interactions, and evades the quadratic complexity which self-attention suffers from.

enable long-range interactions across the image and maintain the computation of linear efficiency, making scaling to high-resolution synthesis easy. Main idea is to iteratively propagate information from a set of latent variables to the evolving visual features and vice versa to support the refinement of each in light of the other.

Fig. 3 (b) shows two types of attention operations over the bipartite graph: simplex and duplex. Simplex attention permits communication in one direction, from the latents to the image features, while duplex attention enables both top-down and bottom up connections between latents and image features. In generateor, it iteratively propagates information between latent components and the image features bidirectionally, to support finer refinement. It can maintain computation of linear efficiency, making scaling to high-resolution synthesis is easy.

The architecture of MorphGANFormer generator is illustrated in Fig. 4. It contains two parts: mapping network and synthesis network. The mapping network is composed of several feed-forward layers that receive a randomly sampled vector Z and output an intermediate vector Z', which in turn interacts directly with each transformer layer through the synthesis network with added noise to modulate the features of the evolving image. Finally, the intermediate vector Z' is transformed into an image X' as the output of the synthesis network.

The latent code Z, has the dimension of 17×32, denoted as [z1, z2, ..., z16, z17], in which [z1, ..., z16] are 16 components of the local-style latent code that are used to interact with the feature of the image through spatial attention, and z17 is a global-style component to globally modulate the feature of the image. The dimension of each component is 32×1. Figs. 1 (a) and (b) show the main difference in latent space between StyleGAN and MorphGANFormer. StyleGAN uses one global monolithic latent to impact the evolving image features of the whole scene uniformly, but in our work, we design a compositional latent space making the latent and image features attend to each other to capture the scene structure.

The synthesis network contains nine stacked synthesis blocks starting from a 4×4 grid and up to produce a final high-resolution image with 1024×1024 resolution. In a synthesis block, the bipartite (duplex) attention operation propagates information from the latent space to the image grid, followed by convolution and upsampling. Gaussian noise is added to each of the activation maps before the attention operations. A different sample of noise is generated for each block and interpreted on the basis of the scaling factors of that layer. The most important part of the synthesis block is the Synthesis Layer. For the first 8 blocks, the Synthesis Layer contains an affine transformation layer (translation, resizing, and rotation), a bias activation layer, and a transformer layer with bipartite attention operation. The blocks 16×16 to 512×512 have the same architecture as the block 8×8 which contains two Synthesis and one Conv2d layer. The Conv2d layer is the convolution layer with optional up-sampling or down-sampling. The last block removes the attention operation and adds an RGB layer to map the dense image features to RGB images.

B. Latent Code Learning

In StyleGAN [7], [22], it uses a latent code to control the style of all features globally. Although it can successfully disentangle global properties, it is more limited in its ability to perform spatial decomposition, as it does not provide a direct means to control the style of localized regions within the generated image. Luckily, the bipartite transformer offers a solution to meet this goal. Instead of controlling the style of all features globally, this attention layer can perform region-wise adaptive modulation. This approach achieves layer-wise decomposition of visual properties, allowing the model to control global aspects of the picture, such as pose, lighting conditions, or color schemes, in a coherent manner over the entire image.

In our method, we use the MorphGANFormer generator that is well trained in a large FFHQ face database [22] with a resolution of 1024×1024 as a basic module to obtain the latent code of the input image. The pipeline is shown in Fig. 2. The pipeline follows a pretty straightforward optimization framework used in [40], [41]. The bipartite attention operation can propagate information from the latent to the image grid, followed by convolution and upsampling. These are stacked multiple times starting from a 4×4 grid and up to 1024×1024 high-resolution images.

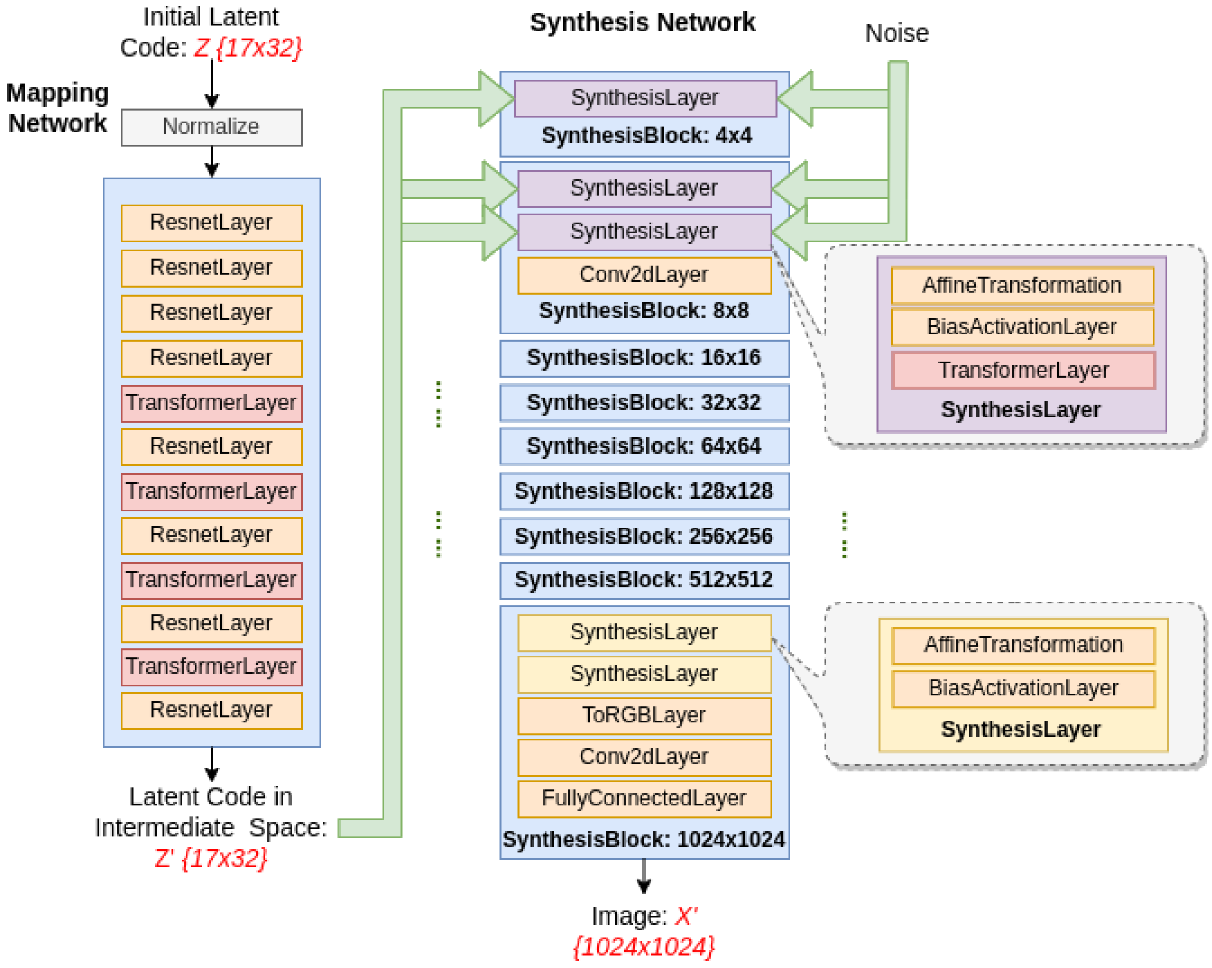


Fig. 4. The architecture of generator G in MorphGANFormer, which contains a mapping network that maps a randomly sampled vector into a intermediate space and a synthesis network that generates a image based on the latent code.

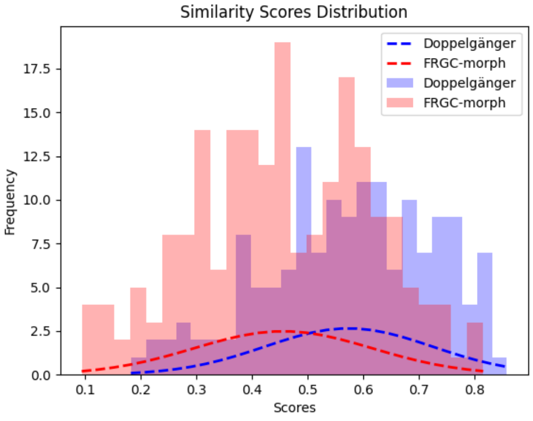


Fig. 5. Similarity score distribution of bona fide pairs on Doppelgänger and FRGC-morph datasets.

C. Loss Functions

To measure the similarity between the input image X and the generated image $G(Z)$ (X') using the learned latent code during optimization, we employ a loss function that is a weighted combination of the Wing Loss [42] based on facial landmarks, the biometric loss based on the distance of matching two faces, VGG-16 perceptual loss [43], and pixel-wise mean square error (MSE):

$$L_{total} = \alpha_1 L_{wing} + \alpha_2 L_{biom} + \alpha_3 L_{percept} + \alpha_4 L_{mse} \quad (1)$$

where α_1, α_2, α_3 and α_4 are weights.

We employ two loss functions related to face content. One is Wing Loss [42], which was originally proposed for facial landmark localization to improve deep neural network training ability for small and medium range errors in sample landmarks. The formula is defined as follows:

$$L_{wing} = \begin{cases} \beta ln(1 + |x|/\epsilon) & if|x| < \beta \\ |x| - C & otherwise \end{cases} \quad (2)$$

where the nonnegative factor β sets the range of the nonlinear part to $(-\beta,\ \beta)$, ϵ limits the curvature of the nonlinear region, $|x|$ means the magnitude of the gradients between the landmark points of $G(Z)$ and X. C = $\beta - \beta ln(1 + \beta/\epsilon)$ is a constant that smoothly links the linear and nonlinear parts defined in part.

The other is biometric loss by calculating the matching distance of the faces. This loss is used to handle the biometric

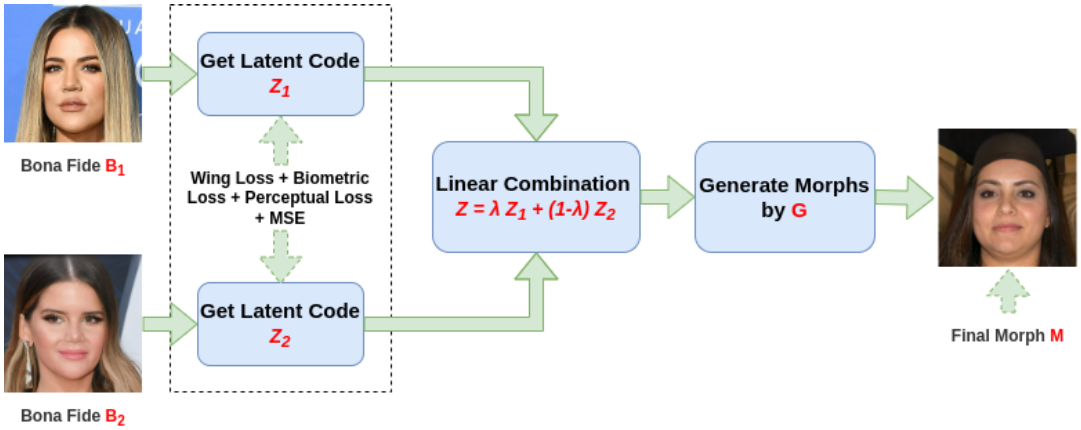


Fig. 6. The pipeline of face morphing.

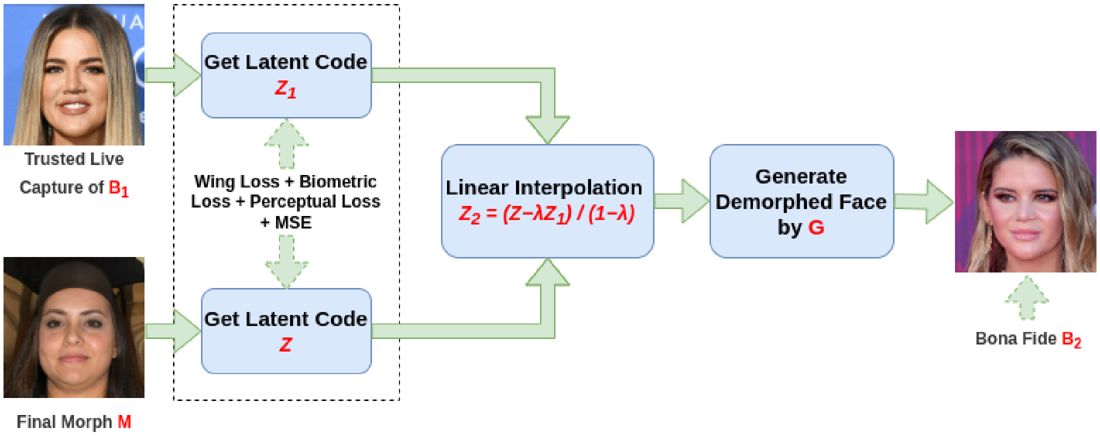

Trusted Live Capture of B1
Get Latent Code Z1
Wing Loss + Biometric Loss + Perceptual Loss + MSE
Get Latent Code Z
Final Morph M
Linear Interpolation Z2 = (Z−λZ1) / (1−λ)
Generate Demorphed Face by G
Bona Fide B2

Fig. 7. The pipeline of face demorphing.

aspect of morphing and to make sure that the morphed faces are related to the original bona fide faces. The matching distance can induce a penalty for the generator during the latent code optimization process if the morphed outputs are not comparable to the original images in terms of biometric utility. The distance between two faces is calculated using the cosine similarity score based on the histogram of oriented gradients (HOG) [23] features, which can be defined as:

$$L_{biom} = 1 - \frac{HOG_{G(Z)} \cdot HOG_X}{\|HOG_{G(Z)}\|\|HOG_X\|}. \tag{3}$$

The study [44], [45] found that the learned filters of the VGG image classification model [46] are excellent general-purpose feature extractors, so they are used to measure the high-level similarity between images perceptually by the covariance statistics of the extracted features, which is formalized as perceptual loss [43]. For the perceptual loss term $L_{percept}$ in Eq. 1, we define it as:

$$L_{percept}(G(Z), X) = \sum_{j=1}^{4} \frac{\lambda_j}{N_j} \|F_j(G(Z)) - F_j(X)\|_2^2 \tag{4}$$

where $G(\cdot)$ is the well trained MorphGANFormer generator, Z is the latent code to optimize, $G(Z)$ is the embedded image, $X \in R^{n \times n \times 3}$ is the target image, N is the number of scalars in the image (i.e., $N = n \times n \times 3$), F_j is the output of the features of the VGG-16 layers conv1_1, conv1_2, conv3_2, and conv4_2, respectively, N_j is the number of scalars in the output of the j-th layer, λ_j is a factor. For the pixel-wise MSE loss term L_{mse}, it is defined as:

$$L_{mse}(G(Z), X) = \frac{1}{N} \|G(Z) - X\|_2^2. \tag{5}$$

The reason for choosing perceptual loss together with pixel-wise MSE loss is that pixel-wise MSE loss alone cannot easily find a high-quality latent vector. Perceptual loss can guide optimization to the right region of the latent space acting as a regularizer.

Given two face images B_1 and B_2, with their respective latent vectors Z_1 and Z_2, face morphing is calculated by linear interpolation:

$$Z = \lambda Z_1 + (1 - \lambda) Z_2, \lambda \in (0, 1) \tag{6}$$

and the final morphing result is generated from the generator G using the latent code Z. The commonly used λ is 0.5.

D. Face Morphing and De-Morphing

Figs. 6 and 7 show the main pipelines of face morphing and demorphing, respectively.

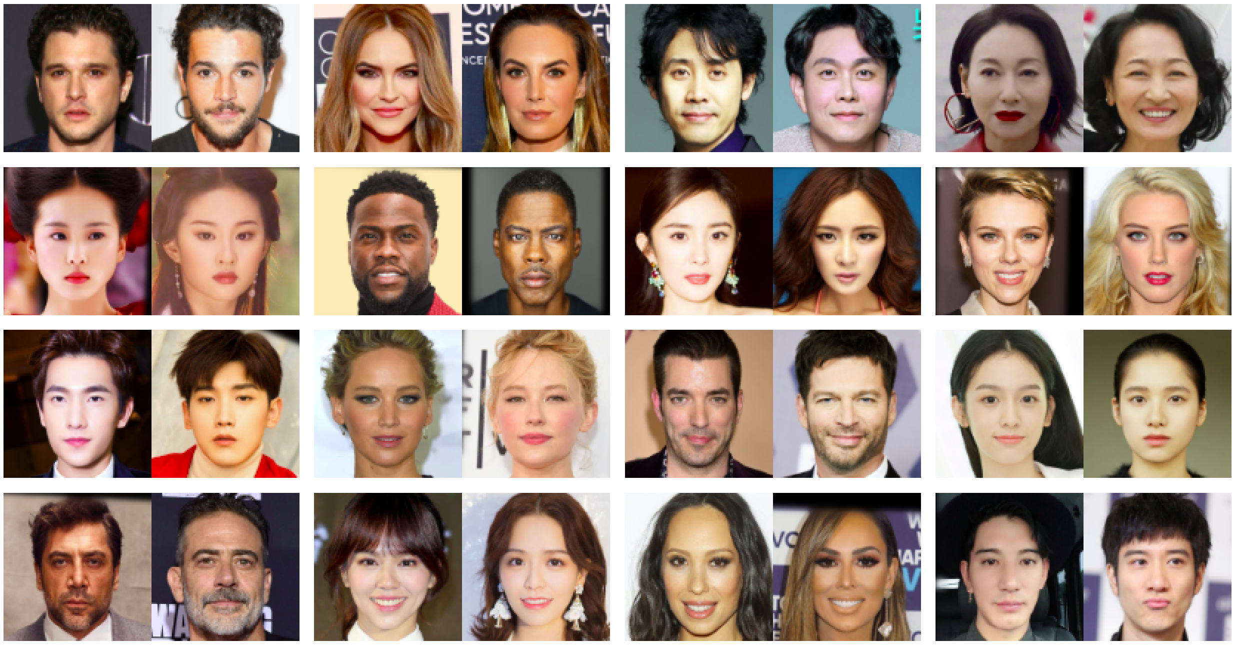


Fig. 8. Some sample pairs of bona-fide face images from the Doppelgänger dataset (note that these look-alike pairs do not have biological connections).

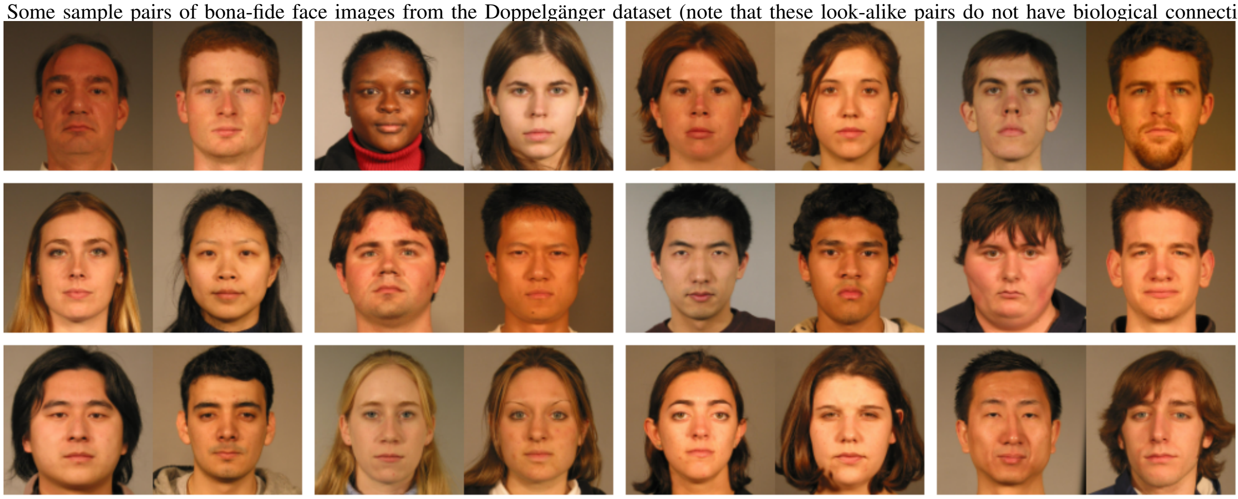


Fig. 9. Some sample pairs of bona-fide face images from the FRGC-morph dataset.

The basic idea of embedding a given image onto the manifold of the pre-trained generator is the following. With an initial latent code Z as the starting point, the model tries to find an optimized latent code Z^* that minimizes the loss function defined to measure the similarity between the target image and the image generated using Z^*. For the initialization of latent codes, we use the mean $\overline{Z}$ of 10,000 latent vectors that are randomly sampled from a uniform distribution of [-1,1], and we expect the optimization to converge to a vector Z^* so that the generated image X' has high similarity to the target image X. We also consider noise-space optimization [47] to complement latent-space embedding, which further improves quality.

The basic idea of demorphing [24] is to try to reverse the morphing process. In the morphing attack, a morphed image can be treated as a linear combination $M = B_1 + B_2$, where B_1 and B_2 are the bona fide faces of two subjects. In a general face verification process without a morphing attack, M can be treated as a combination of two identical face images of one person. In the morphing attack situation, during the face verification process, the system receives $\hat{B}_1$, a live captured variant of B_1, and the demorphing task is to calculate the demorphed image $\hat{B}2$ by removing $\hat{B}1$ from M, which is $\hat{B}2 = M - \hat{B}1$.

Given the live trusted capture of one bona fide face image B_1 and the morphed face image M, with their respective latent vectors Z_1 and Z, face demorphing is calculated in latent space by:

$$Z_2 = \frac{Z - \lambda Z_1}{(1-\lambda)}, \lambda \in (0,1) \tag{7}$$

and final demorphing result is generated from the generator G using the latent code Z_2.

IV. Experimental Setup

A. Database Description

Table I presents the database used in our experiment: the newly constructed Doppelgänger face morphing database and

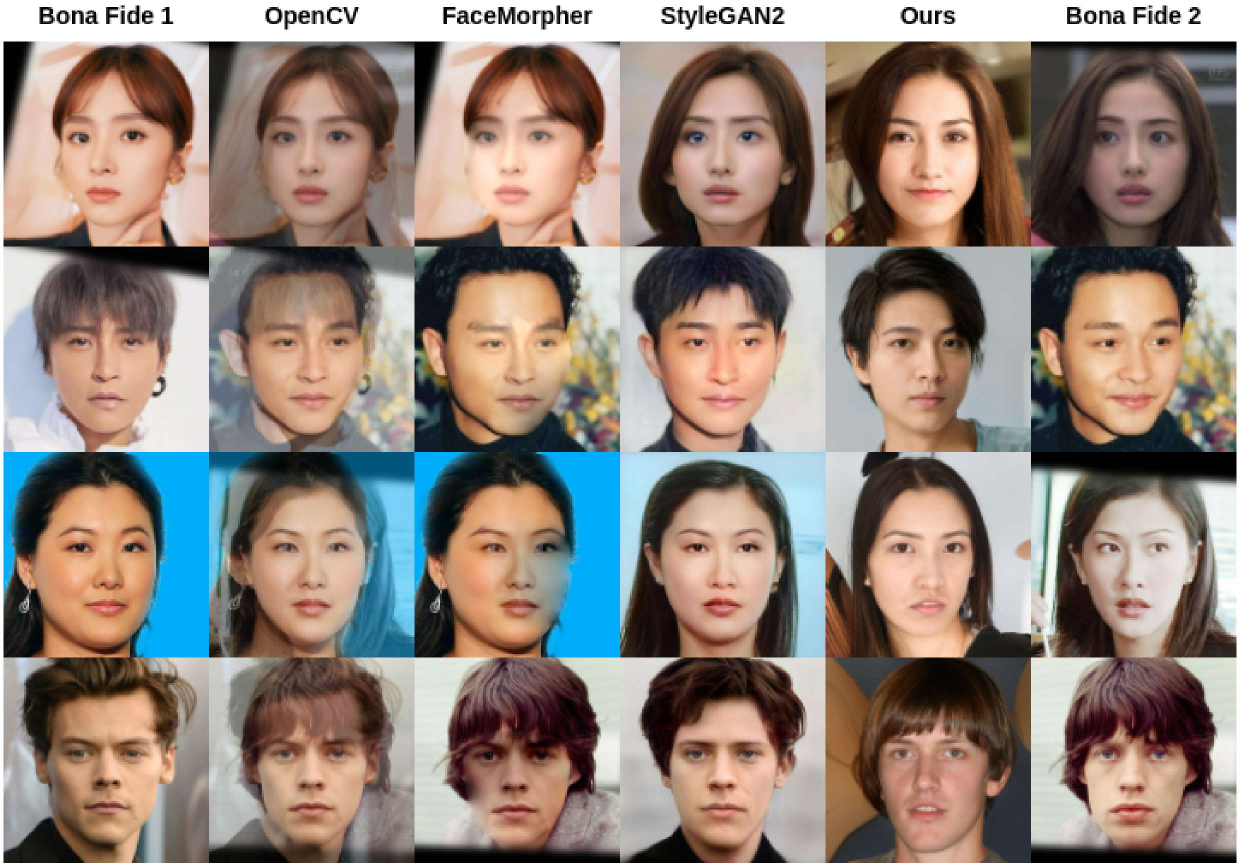


Fig. 10. Face morphing results in the Doppelgänger Morphs database without any post-processing.

TABLE I
THE DATA USED IN OUR EXPERIMENT. ONE IS THE NEWLY CONSTRUCTED DOPPELGÄNGER FACE MORPHING DATABASE AND THE OTHER ONE IS RECONSTRUCTED FRGC-MORPH DATASET.

Database	Subset	#Number	Resolution
Doppelgänger	bona fide	153 pairs	1024x1024
	trusted live captures	306	1024x1024
	FaceMorpher	150	1024x1024
	OpenCV	153	1024x1024
	StyleGAN2	153	1024x1024
	MorphGANFormer	153	1024x1024
FRGC-morph	bona fide	204 pairs	1024x1024
	trusted live captures	408	1024x1024
	FaceMorpher	204	1024x1024
	OpenCV	204	1024x1024
	StyleGAN2	204	1024x1024
	MorphGANFormer	204	1024x1024

reconstructed FRGC-morph dataset. Both are composed of bona fide faces, corresponding trusted live captures, four types of morphing results via OpenCV, FaceMorpher, StyleGAN2 and our MorphGANFormer.

Figs. 8 and 9 shows some pairs of bona fide face images from Doppelgänger and FRGC-morph dataset. Note that for the former we are guaranteed that the pair will look similar; for the latter, we have adopted a strategy of random pairing so the likelihood of obtaining two similar bona fide images is low.

We use the real images in two databases as bona fide faces. The first is the Doppelgänger dataset in which a name-pair list is created to gather the faces of celebrities that look alike, with the same gender and ethnicity. All faces are rotated to align the eyes on a horizontal line. Only one image per identity is considered. Finally, we obtained 153 pairs (95 female; 58 male) with the size of 1024×1024 resolution. The second dataset is constructed from FRGC [48]. All faces are cropped, aligned, and resized to 1024×1024 resolution. Subjects with the same gender are randomly selected to compose bona fide pairs for face morphing. Each subject is selected only once. Finally, we get 204 pairs (112 male and 92 female). For both datasets, we obtain one extra image for each subject as a trusted live capture for de-morphing task. Fig. 5 illustrates the different distributions of similarity scores between two bona fide faces per pair in the Doppelgänger and FRGC-morph datasets using FaceNet [49] feature, which shows that the Doppelgänger pairs have higher similarity scores than the FRGC-morph.

B. Experimental Setup

For the latent code initialization, we use the mean $\overline{Z}$ of 10,000 latent vectors that are randomly sampled from a uniform distribution of [-1,1]. For perceptual loss, we choose pre-trained VGG-16 as the backbone network to extract image feature. For Wing loss, we use dlib toolbox [28] to detect 68 facial points for calculation. For the distance between the two

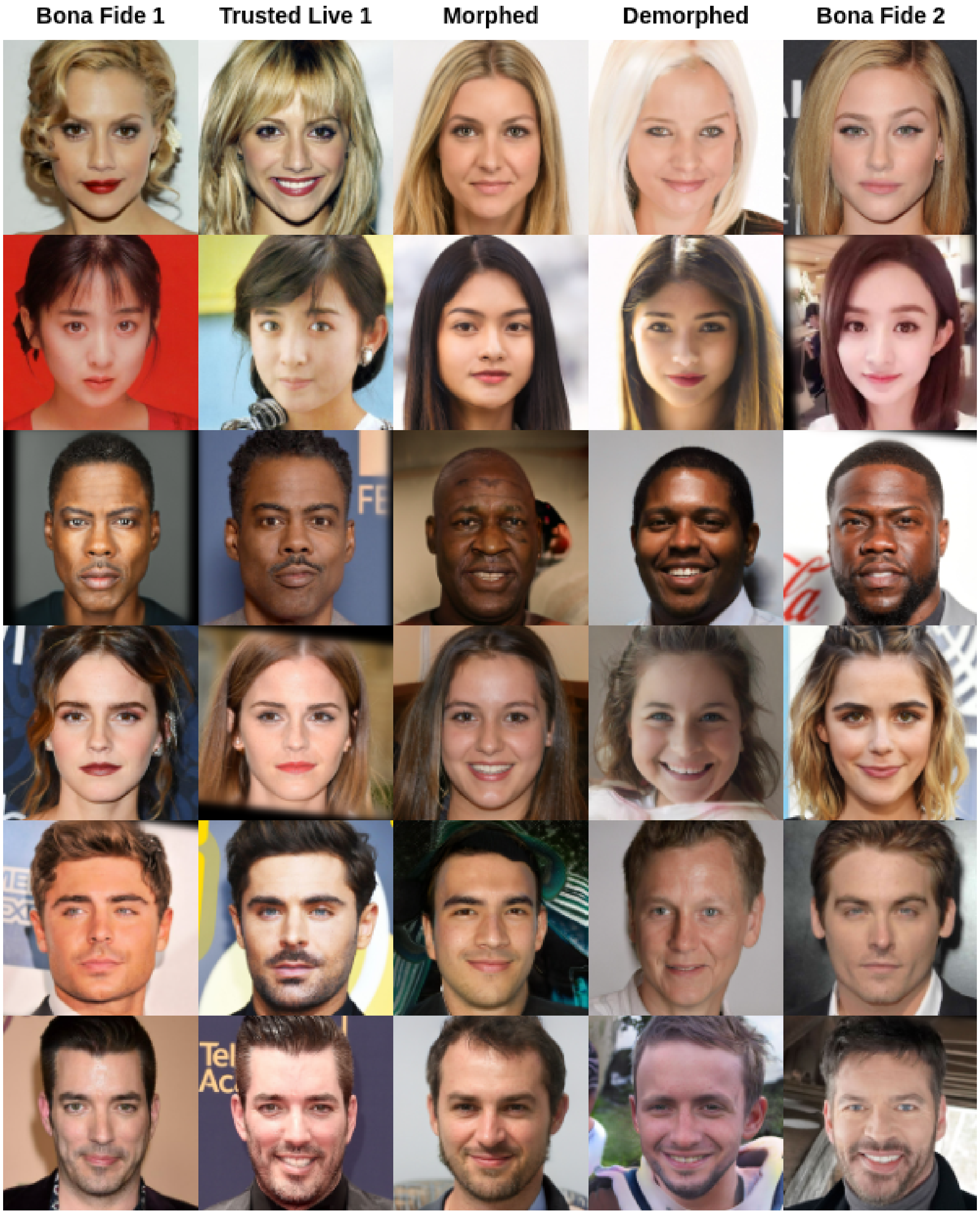


Fig. 11. Some demorphed results on Doppelgänger dataset.

faces, we use HOG feature [23] of the faces to calculate the matching score. We use Adam optimizer with a learning rate of 0.01 to optimize the latent code learning procedure with α_1=0.02, α_2=1.0, α_3=1.0, and α_4=1.0 for loss functions. We set 1,500 gradient descent steps for the optimization, and keep the latent code with the lowest loss value for generation.

C. Vulnerability Test

We evaluate the vulnerability of three face recognition models to the morphing attacks created by our morphing framework. ArcFace [52] introduced Additive Angular Margin

TABLE II
MMPMR (%) ON DOPPELGÄNGER AND FRGC-MORPH DATABASE.

Dataset	Morph Type	ArcFace	FaceNet	LBP
Doppelgänger	OpenCV [2]	94.73	82.23	87.50
	FaceMorpher [3]	81.21	73.83	87.92
	StyleGAN2 [7]	84.21	70.65	85.52
	MorphGANFormer	90.08	70.92	89.77
FRGC-morph	OpenCV [2]	87.75	74.51	94.61
	FaceMorpher [3]	80.39	72.06	85.78
	StyleGAN2 [7]	38.73	35.78	78.43
	MorphGANFormer	48.04	42.65	84.80

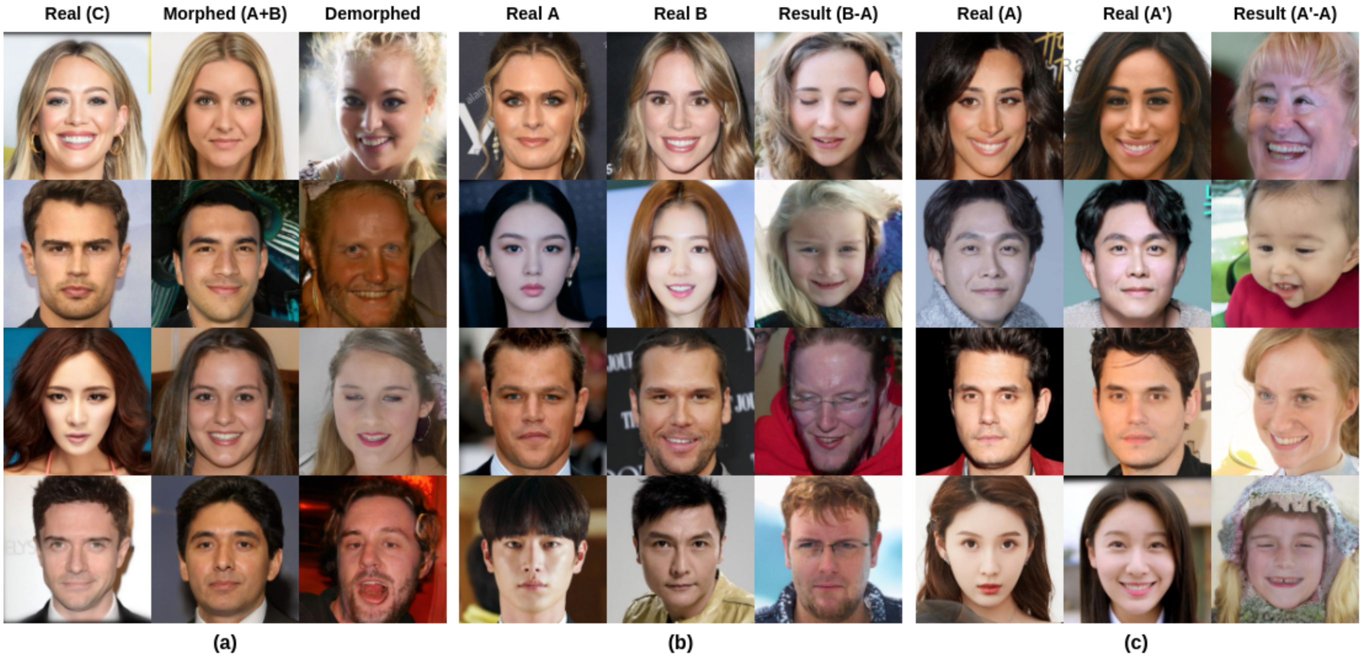


Fig. 12. Some demorphing results using different inputs on Doppelgänger dataset. (a) The inputs are morphed faces combined by identity A and B, and trusted live captures of identity C; (b) The inputs are real faces of identity B as morphed images, and real faces of identity A as trusted live captures; (c) The inputs are real faces A' as morphed images, and the other real faces A of the same identity as trusted live captures.

TABLE III
MMPMR (%) WITH ABLATION STUDY ON DOPPELGÄNGER DATABASE.

Loss	ArcFace	FaceNet	LBP
$Biom_{FaceNet}$	56.58	50.53	82.11
$Biom_{ArcFace}$	53.29	47.24	80.79
$Biom_{LBP}$	50.66	43.95	90.00
$Biom_{HOG}$	77.63	45.92	86.71
Percept	53.29	43.95	78.82
Percept+Wing	82.24	59.08	88.68
Percept+Wing+MSE	84.87	62.37	89.34
$Biom_{HOG}$+Percept	86.18	59.74	88.03
$Biom_{HOG}$+Percept+Wing	85.53	61.05	88.03
$Biom_{HOG}$+Percept+Wing+MSE	90.08	70.92	89.77

loss to improve the discriminative ability of the face recognition model. It scored state-of-the-art performance on several face recognition evaluation benchmarks such as Labeled Faces in the Wild (LFW) [56] 99.83% and YouTube Face (YTF) [57] 98.02%. We use an ArcFace model based on ReseNet-100 [58] architecture pre-trained on a refined version of the MS-Celeb-1M dataset (MS1MV2) [59] to extract face features. FaceNet [49] directly learns an embedding mapped from input to an Euclidean space in which the Euclidean distance indicates the similarity of the face. It uses triplets of tightly cropped face patches generated by an online triplet mining method to train the network, and its output is a compact 128-D embedding. Local Binary Pattern (LBP) [60] is a hand-crafted feature that describes the texture characteristics of surfaces. By applying LBP, the probability of the texture pattern can be summarized into a histogram. It is a commonly used feature in face recognition domain.

Dlib face detector [28] is used to segment the face region. The cropped face is normalized according to the eye coordinates and resized to a fixed size of 224×224 pixels. The single feature extraction (ArcFace, FaceNet, and LBP) procedure is performed on the processed faces. Ideally, a strong morphing attack will have a similar and high similarity score to the target identities. We present the vulnerability results in a quantifiable manner by giving the Mated Morphed Presentation Match Rate (MMPMR) [27] based on the decision threshold at the false match rate (FMR) of 0.1%. Note that all vulnerability results are presented on the testing data.

Table II shows the MMPMR (%) values of different morphing methods using ArcFace, FaceNet and LBP features. And Fig. 10 shows some morphing samples in the Doppelgänger database. We can see that for landmark-based morphing attacks, like OpenCV and FaceMorpher, it has high MMPMR values, indicating it highly preserves the characteristic of both bona fide identities, but the image artifacts caused by blending on image level are obvious too. In contrast, GAN-based morphing methods improve the visual quality of morphed images. However, synthetic-like generation artifacts, as shown in the StyleGAN2 attack, make morphing faces less realistic and natural. Our model has the same or even better ability to preserve the facial identities as landmark-based models and can also generate visually realistic and natural faces.

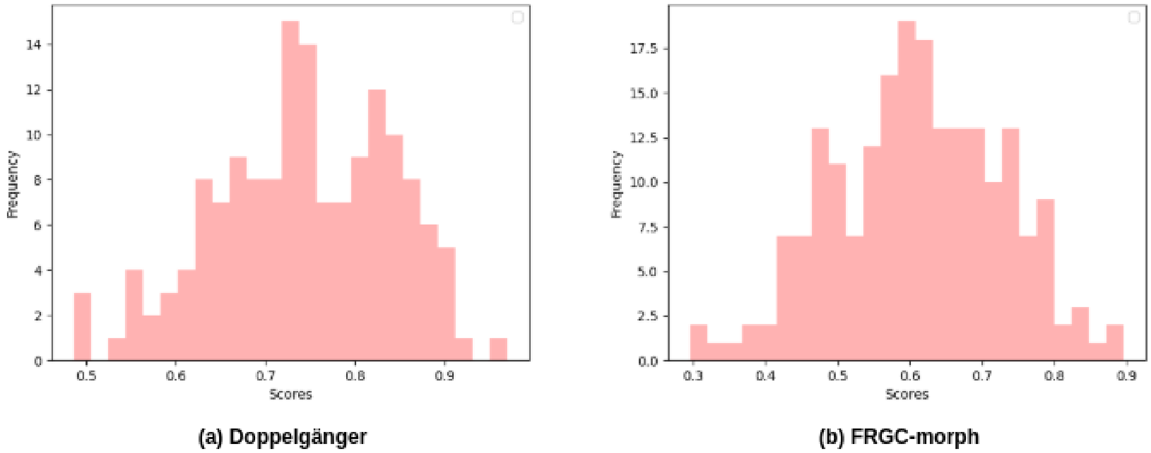
We also did an ablation study with different loss functions on Doppelgänger dataset as shown in Table III. The first part shows some results using different facial features to calculate the face matching distance. From the second and third parts, we can see that, with the combination of more loss functions, the MMPMR value increases.

D. Detectability Analysis

To thoroughly evaluate the detectability of MorphGAN-Former attacks, we selected several popular methods used in

TABLE IV
PERFORMANCE (%) COMPARISON OF MAD ON OPENCV, FACEMORPHER, STYLEGAN2, AND OUR METHOD. ACCU. - ACCURACY.

		OpenCV [2]			FaceMorpher [3]			StyleGAN2 [7]			**MorphGANFormer**		
Dataset	MAD Method	Accu.	D-EER	ACER	Accu.	D-EER	ACER	Accu.	D-EER	ACER	Accu.	D-EER	ACER
Doppelgänger	MobileNetV2 [50]	66.45	36.18	49.50	66.00	42.36	50.82	66.45	37.50	49.51	65.57	59.87	50.82
	NasNetMobile [51]	68.64	35.53	43.42	65.12	45.02	49.26	62.50	45.56	52.63	61.84	65.13	53.62
	ArcFace [52]	66.23	40.13	40.79	62.91	46.35	46.11	59.43	46.88	50.99	58.77	51.97	51.97
	MB-LBP [53]	66.67	44.24	47.53	67.99	43.02	46.09	67.11	45.39	46.88	64.47	51.32	50.82
	FS-SPN [54]	48.68	44.74	43.59	45.47	47.67	48.15	50.00	42.11	41.61	44.96	50.66	49.18
	MixFaceNet-MAD [55]	67.76	34.21	33.55	63.36	39.54	40.31	57.02	50.66	49.67	57.89	46.71	48.36
FRGC-morph	MobileNetV2 [50]	44.28	28.43	42.16	44.12	36.27	42.40	44.77	18.26	41.42	33.33	57.35	58.58
	NasNetMobile [51]	71.57	29.53	32.60	69.93	32.84	35.05	68.46	33.82	37.25	59.48	49.02	50.74
	ArcFace [52]	66.34	43.63	46.94	65.36	44.73	48.41	66.67	37.25	46.45	68.79	38.73	43.26
	MB-LBP [53]	67.16	43.75	46.57	66.67	42.65	47.30	63.73	51.72	54.90	66.50	49.02	47.55
	FS-SPN [54]	55.72	46.57	47.43	54.90	47.06	48.65	72.06	24.02	22.92	58.33	45.59	43.50
	MixFaceNet-MAD [55]	67.48	33.33	39.71	65.52	40.20	42.65	62.91	44.12	46.57	61.11	49.02	49.26



(a) Doppelgänger

(b) FRGC-morph

Fig. 13. Similarity score distribution between restored faces and real faces of the bona fide on (a) Doppelgänger and (b) FRGC-morph datasets based on FaceNet feature.

face recognition [52], pre-trained deep models [50], [51], [61], [62] on ImageNet [63], and existing morphing attack detection methods [53]–[55], [64], [65], for comparison. We measure the attack detection performance on our generated attacks, and other types of attacks, like OpenCV [2], FaceMorpher [3], and StyleGAN2 [7], based on the bona fide faces in Doppelgänger and FRGC-morph databases.

We evaluate the detectability of our attacks as unknown attacks, i.e., novel attacks unknown to the detection algorithm. In this case, the training data come from the attacks of LMA [4], WebMorph [5], AMSL [32], MorGAN [4] and CIEMorGAN [33] attacks introduced in [66], and their corresponding bona fide faces, which contains 1,838 images (bona fide: 918; morphed: 920) in total. The test data are from Doppelgänger (153 morphed + 306 bona fide) and FRGC-morph (204 morphed + 408 bona fide) datasets, respectively. We trained a binary classifier using the training data. After the detector is well trained, it is used to predict bona fide and our MorphGANFormer attacks (or OpenCV [2], FaceMorpher [3], StyleGAN2 [7] attacks).

Following previous morphing attacks detection (MAD) studies [67], [68], we report performance using accuracy, D-EER, and ACER. Detection Equal-Error-Rate(D-EER) is the error rate for which both BPCER and APCER are identical. The average classification error rate (ACER) is calculated by the mean of the APCER and BPCER values. The attack presentation classification error rate (APCER) reports the proportion of morph attack samples incorrectly classified as bona fide presentation, and the Bona Fide Presentation Classification Error Rate (BPCER) refers to the proportion of bona fide samples incorrectly classified as morphed samples. The results are shown in Table IV. Compared to the OpenCV, FaceMorpher, and StyleGAN attacks, the MorphGANFormer attacks are more challenging. Unlike vulnerability, we note that the detectability performance gap between the Doppelgänger and FRGC datasets is small.

TABLE V
DEMORPHING ACCURACY (%) ON DOPPELGÄNGER AND FRGC-MORPH.

	ArcFace	FaceNet	LBP
Doppelgänger Pairs	54.90	62.75	88.24
FRGC-morph Pairs	29.94	37.25	68.14

E. Performance of De-morphing

To quantitatively evaluate the performance of the demorphing result, ArcFace, FaceNet, and LBP are adopted to compare the restored facial image $\hat{B_2}$ with B_2 and B_1, respectively. When the system determines that $\hat{B_2}$ matches B_2, but does not match B_1, the demorphing is considered successful. We use a restoration accuracy introduced in FD-GAN [34] as a measure metric to check the demorphing performance. In our paper, we termed restoration accuracy as demorphing accuracy.

The demorphing accuracy is defined as the percentage of the number of successfully demorphed facial images in the total number of demorphed facial images. The decision threshold for similarity scores is set as the value of the false match rate (FMR) at 0.1%. Table V shows the result.

Fig. 11 shows some results of face demorphing on Doppelgänger dataset. We use morphed face and one trusted live capture of bona fide 1 to restore the face of bona fide 2, as shown in column 'Demorphed'. It can be clearly seen that demorphed image has a good resemblance to the face of bona fide 2, justifying the effectiveness of our defense strategy in the latent space.

Fig. 12 shows some results using randomly selected inputs to do demorphing. Fig. 12 (a) uses a morphed face generated by bona fide A and B, and the trusted live capture from a third identity C, as input. Fig. 12 (b) uses a real face image of identity B as morphed face to be input to the demorphing model, and the other real face image of identity A as the trusted live capture. Fig. 12 (c) applies two face images of the same identity as inputs. The demorphed results are various and uncontrollable with low quality. Obvious artifacts can be easily spotted.

Fig. 13 presents the similarity scores distribution between the demorphed faces of bona fide 2 and real faces of bona fide 2 on two datasets based on FaceNet feature. It can be seen that demorphing can achieve reasonably good matching scores on both datasets, implying the detectability of our defense strategy in the latent space. Between Doppelganger and FRGC, we observe that FRGC has lower matching scores than Doppelganger, suggesting less vulnerability. The choices of bona fide pair for face morphing, which is related to the trade-off between detectability and vulnerability, deserves further systematic study.

V. Conclusion and Future Work

Face morphing attacks have received increasing attention in recent years. Generation approaches such as GAN-based are among the leading techniques. However, existing methods suffer from noticeable blurring and synthetic-like generation artifacts. In this paper, we designed a transformer-based alternative to face morphing, which demonstrated its superiority to StyleGAN-based methods. Four particular loss functions were employed to maximize the similarity between the generated face image and the target face image. We also extended the study of transformer-based face morphing to demorphing, the dual operation. Future work includes an improved understanding of the trade-off between vulnerability and detectability as well as other morphing approaches such as diffusion models [69].

References

[1] S. Venkatesh, R. Ramachandra, K. Raja, and C. Busch, "Face morphing attack generation and detection: A comprehensive survey," *IEEE transactions on technology and society*, vol. 2, no. 3, pp. 128–145, 2021.

[2] "Opencv," https://learnopencv.com/face-morph-using-opencv-cpp-python/, accessed: August 2021.

[3] "Facemorpher," https://github.com/yaopang/FaceMorpher, accessed: August 2021.

[4] N. Damer, A. M. Saladie, 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).* IEEE, 2018, pp. 1–10.

[5] L. DeBruine, "Webmorph morphing tool," 2016.

[6] 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.

[7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision.* Springer, 2020, pp. 213–229.

[12] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[13] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1833–1844.

[14] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 040–14 049.

[15] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *European Conference on Computer Vision.* Springer, 2020, pp. 528–543.

[16] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "Styleswin: Transformer-based gan for high-resolution image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 304–11 314.

[17] D. A. Hudson and L. Zitnick, "Generative adversarial transformers," in *International conference on machine learning.* PMLR, 2021, pp. 4487–4499.

[18] D. Arad Hudson and L. Zitnick, "Compositional transformers for scene generation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9506–9520, 2021.

[19] L. Zhao, Z. Zhang, T. Chen, D. Metaxas, and H. Zhang, "Improved transformer for high-resolution gans," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 367–18 380, 2021.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[21] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 745–14 758, 2021.

[22] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[24] M. Ferrara, A. Franco, and D. Maltoni, "Face demorphing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 1008–1017, 2017.

[25] ——, "Face demorphing in the presence of facial appearance variations," in *2018 26th European Signal Processing Conference (EUSIPCO).* IEEE, 2018, pp. 2365–2369.

[26] A. Röttcher, U. Scherhag, and C. Busch, "Finding the suitable doppelgänger for a face morphing attack," in *2020 IEEE international joint conference on biometrics (IJCB)*. IEEE, 2020, pp. 1–7.

[27] U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R. N. Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel *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)*. IEEE, 2017, pp. 1–7.

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

[29] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[30] S. Milborrow and F. Nicolls, "Active shape models with sift descriptors and mars," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2. IEEE, 2014, pp. 380–387.

[31] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[32] AMSL, "Amsl face morph image data set," 2021.

[33] N. Damer, F. Boutros, A. M. Saladie, F. Kirchbuchner, and A. Kuijper, "Realistic dreams: Cascaded enhancement of gan-generated images with an example in face morphing attacks," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–10.

[34] F. Peng, L.-B. Zhang, and M. Long, "Fd-gan: Face de-morphing generative adversarial network for restoring accomplice's facial image," *IEEE Access*, vol. 7, pp. 75 122–75 131, 2019.

[35] S. Banerjee and A. Ross, "Conditional identity disentanglement for differential face morph detection," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2021, pp. 1–8.

[36] S. Banerjee, P. Jaiswal, and A. Ross, "Facial de-morphing: Extracting component faces from a single morph," *arXiv preprint arXiv:2209.02933*, 2022.

[37] D. Ortega-Delcampo, C. Conde, D. Palacios-Alonso, and E. Cabello, "Border control morphing attack detection with a convolutional neural network de-morphing approach," *IEEE Access*, vol. 8, pp. 92 301–92 313, 2020.

[38] E. Shiqerukaj, C. Rathgeb, J. Merkle, P. Drozdowski, and B. Tams, "Fusion of face demorphing and deep face representations for differential morphing attack detection," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2022, pp. 1–5.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[40] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 7, pp. 1967–1974, 2018.

[41] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4432–4441.

[42] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2235–2245.

[43] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[44] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[46] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 2015, pp. 730–734.

[47] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan++: How to edit the embedded images?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8296–8305.

[48] "Face recognition grand challenge (frgc)," https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc, accessed: August 2021.

[49] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[51] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[52] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[53] 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.

[54] L.-B. Zhang, F. Peng, and M. Long, "Face morphing detection using fourier spectrum of sensor pattern noise," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[55] N. Damer, C. A. F. López, M. Fang, N. Spiller, M. V. Pham, and F. Boutros, "Privacy-friendly synthetic data for the development of face morphing attack detectors," *arXiv preprint arXiv:2203.06691*, 2022.

[56] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[57] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition, Conf. on*. IEEE, 2011, pp. 529–534.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[59] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European conference on computer vision*. Springer, 2016, pp. 87–102.

[60] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[61] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[62] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[64] L. Debiasi, C. Rathgeb, U. Scherhag, A. Uhl, and C. Busch, "Prnu variance analysis for morphed face image detection," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–9.

[65] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *arXiv preprint arXiv:1808.08396*, 2018.

[66] N. Zhang, S. Jia, S. Lyu, and X. Li, "Fusion-based few-shot morphing attack detection and fingerprinting," *arXiv preprint arXiv:2210.15510*, 2022.

[67] K. Raja, M. Ferrara, A. Franco, L. Spreeuwers, I. Batskos, F. de Wit, M. Gomez-Barrero, U. Scherhag, D. Fischer, S. K. Venkatesh *et al.*, "Morphing attack detection-database, evaluation platform, and benchmarking," *IEEE transactions on information forensics and security*, vol. 16, pp. 4336–4351, 2020.

[68] 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.

[69] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.