2022

Landmark Enforcement and Principal Component Analysis for Improving GAN-Based Morphing

Samuel W. Price
swp0001@mix.wvu.edu

LANDMARK ENFORCEMENT AND PRINCIPAL COMPONENT ANALYSIS FOR IMPROVING GAN-BASED MORPHING

Samuel Price

Thesis submitted to the Benjamin M. Statler College of Engineering
and Mineral Resources at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering

Nasser Nasrabadi, Ph.D., Chair
Jeremy Dawson, Ph.D.
Matthew Valenti, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2022

Keywords: Face Recognition, Morph Attack, Morph Generation, Deep Learning, Principal Component Analysis

ABSTRACT

Landmark Enforcement and Principal Component Analysis for Improving GAN-Based Morphing

Samuel Price

Facial Recognition Systems (FRSs) are a key target for adversaries determined to circumvent security checkpoints. Morph images threaten FRS by presenting as multiple individuals, allowing an adversary to swap identities with another subject. Although morph generation using generative adversarial networks (GANs) results in high-quality morphs without possessing the spatial artifacts caused by landmark-based methods, there is an apparent loss in identity with standard GAN-based morphing methods. In this thesis, we examine landmark-based and GAN-based morphing methods to fuse the advantages of both methodologies. We propose a novel StyleGAN2 morph generation technique by introducing a landmark enforcement method. Considering this method, we aim to enforce the landmarks of the morph image to represent the spatial average of the landmarks of the bona fide faces.

Loss in visual quality of images projected into the latent space of the StyleGAN2 model reduces the potential quality of the morphs. We compare previous image inversion methods to derive a novel method to improve the latent space representation of an image. To further improve the perceptual quality of the morphs, we examine the noise inputs of our model. Trainability of the noise input is evaluated to learn reconstruction information the latent codes cannot represent. Further exploration of the latent space of our model is conducted using Principal Component Analysis (PCA) to pronounce the effect of the bona fide faces on the morphed latent representation. This work's contributions include a novel GAN-based morphing method to attack FRS at higher success rates than alternative GAN-based methods. We improve image inversion into the latent space by exploring the model's noise input while enforcing the balance of latent identities through PCA.

ACKNOWLEDGEMENTS

There are numerous people I would like to thank as I could not have completed this work without them.

I would first like to thank Dr. Nasrabadi for the guidance, knowledge, and confidence he shared with me from the very beginning. I would also like to thank Dr. Dawson and Dr. Valenti for their time and influence on this work. I want thank Sobhan Soleymani, Kelsey O'Haire, and Baaria Chaudhary for their contributions to this work as well. Finally, I would like to thank my loving family for their unwavering support of my academic pursuits as I could not have done this without their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Generative Adversarial Networks [5] (GAN) continue to grow in popularity in areas such as deepfake generation: realistic images generated by a deep neural network (DNN) [6]. With recent improvements in the realistic generation abilities of GANs [2, 3], the threat deepfakes pose to personal reputation, corporate sabotage, and national security grow increasingly concerning [7]. As such, attacks on Facial Recognition Systems (FRS) mount as they continue to serve an integral part of national security, law enforcement, and numerous personally owned devices to verify identity [8, 9]. Border security is a key target as facial recognition is the only biometric required in electronic Machine-Readable Travel Documents (eMRTD) approved by the International Civil Aviation Commission [10]. Deepfakes can attack the enrollment stage of the biometric system integration guideline set by the ICAO by passing two

safeguards: image tampering detection and identity verification. If a deepfake fools both the detector and is identified as the individual in question, a bad actor could slip right through these security measures [11]. Our proposed technique generates a type of deepfake known as a morph that possesses the identity of two individuals capable of fooling both human inspectors and FRS using a GAN.

Facial morph images have proven a threat to FRS when submitted by a bad actor as a means to identify themselves [11]. A facial morph is an artificial face image generated by blending two or more real face images of different individuals. Good facial morphs balance the identities of each real face image used during generation. The contributing subjects can use the morph for verification as FRS would find their identities indistinguishable to that of the morph. A bad actor under scrutiny could find a look-alike individual, morph their faces, and use the resultant morph to pass themselves off as their look-alike. We explore the threat GAN-based morphing poses to FRS as well as improve upon current GAN-based morphing techniques.

1.2 Goals and Objectives

In this thesis, we build upon the works of [3, 4, 12] to generate a face morphing technique utilizing StyleGAN2. Compared to other face morphing techniques, GAN-based face morphing falls short when used to attack FRS [12, 13]. Improvements to early face generating GANs made by Karras *et al.* [2] have increased their threat to FRS due to increased image resolution size and visual quality. Regardless, GAN-based morphs continue to perform significantly worse than alternative techniques such as landmark-based morphing [1]. Whereas landmark-based morphs are generated in the

image domain, GAN-based morphs occurs in the latent space by blending the latent representation of the original facial images. Our technique augments the calculation of the latent representations by blending the landmarks of the original facial images before finding their respective latent representations. This work shows how morphing the landmarks before inverting the bona fide faces into their latent representation allows the morphed face's landmarks to be equidistant from the original subjects', increasing their threat to FRS.

To calculate the latent representations of our bona fide subjects, we derive a loss function from [3, 4, 14] to improve the preservation of identity. Our loss function includes perceptual loss using a DNN to extract features from the bona fide and synthesized images, pixel-wise loss, and regularization losses to improve the quality of the latent representation. Then, we explore alternatives to latent averaging to further improve the quality of the morph image using Principal Component Analysis (PCA). We explore the effects of element-wise and vector-wise selection to blend the latent representations for morphing (see Figure 1.1). With our novel technique, we generate GAN-based morph images which fool FRS at an increased rate while maintaining the image quality to fool human inspectors.

1.3 Thesis Contributions

In this work, we explore the StyleGAN2 [3] architecture to develop a novel technique for image morphing. We evaluate our technique against current landmark-based and GAN-based methods. Therefore, major contributions of this work are as follows:

- An exploration of current landmark-based and GAN-based morph generation methods.
- We introduce a novel GAN-based morph generation method enforcing landmarks.
- An in-depth study into the noise input of the StyleGAN2 model to improve inverted image quality.
- We explore alternative methods to blend latent representations to generate higher quality morphs.
- An evaluation of morphs generated using our novel technique and current GAN-based morphing techniques.

1.4 Thesis Organization

The organization of this thesis are as follows:

- Chapter 2 reviews previous work related to morph generation and the threat they pose to FRS. After discussing FRS used in this work, we discuss landmark-based and GAN-based morphing techniques followed by a review of StyleGAN2 [3] and image inversion.
- Chapter 3 discusses experiments made on GAN-based morphing techniques using StyleGAN2 [3]. We compare previous GAN-based morphing techniques to the new landmark-enforced morphs to evaluate their threat to FRS. Additionally, experiments on applying texture to the generated images and their effect on FRS are discussed.
- Chapter 4 explores alternative latent representation blending to improve the balance the identities of both subjects present in the morph.
- Chapter 5 summarizes our contributions and outlines future work to build upon our techniques.

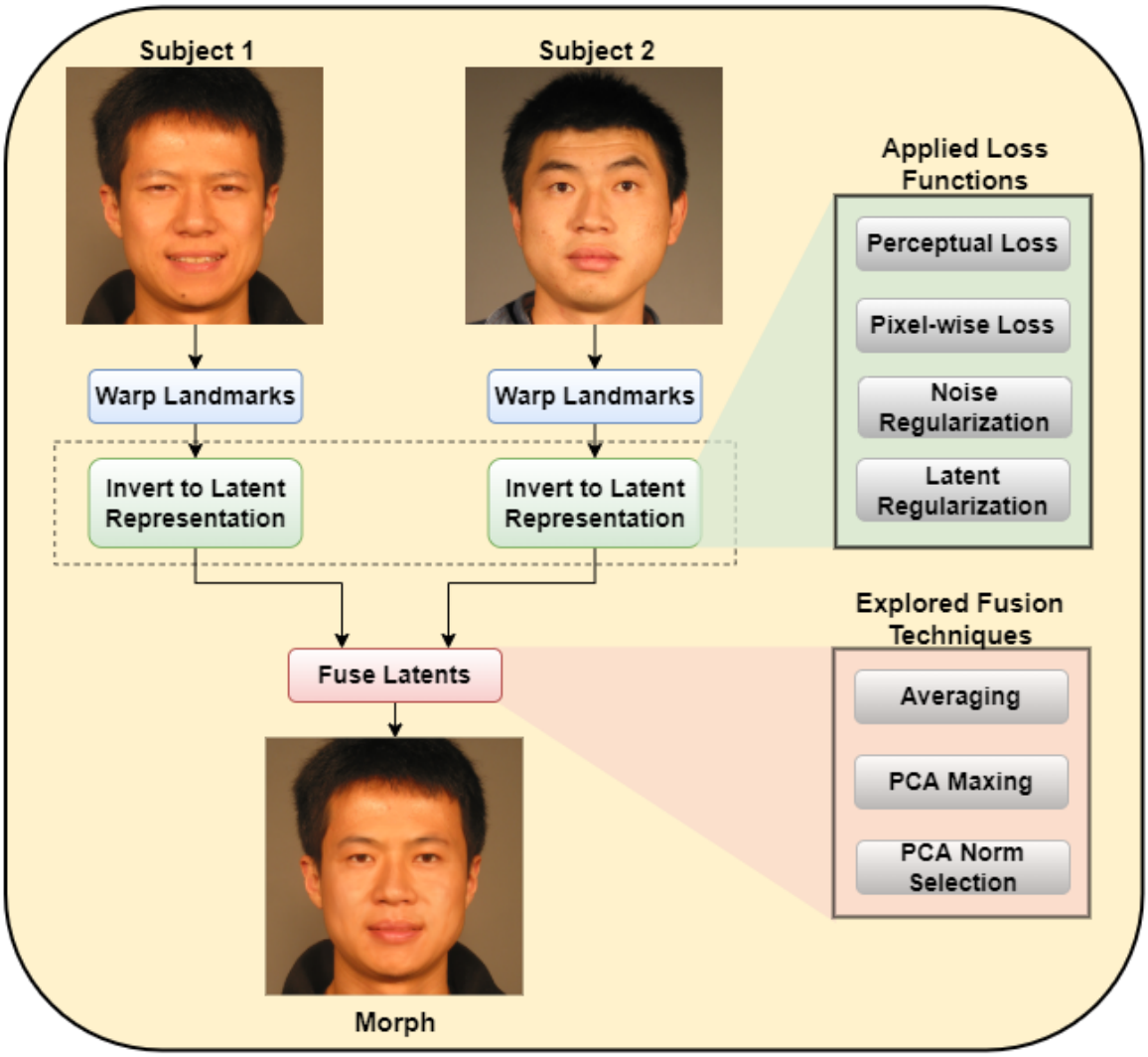

Figure 1.1: Our proposed morphing technique utilizing landmark warping and latent representation averaging.

Chapter 2

Literature Review

2.1 Morph Threat to Facial Recognition Systems

Biometrics are used in areas such as law enforcement, security, and every day conveniences in increasing frequency as a means to identify individuals [8, 9]. Although fingerprints have been used for over a century in law enforcement, fingerprints are not ideal in situations where collecting scans or imprints are not possible, impractical, or too invasive [15]. Facial recognition is the most widely used method of biometric identification. Although this ability is one most humans take advantage of, we are unable to identify faces which we have not seen. Automated approaches are necessary to streamline recognition of faces while improving the accuracy compared to the abilities of a human.

2.1.1 Facial Recognition Systems

In this thesis, two deep face recognition systems were used: FaceNet and ArcFace [16, 17]. Both networks operate as feature extractors which output an embedded vector representation of the inputted image. The embeddings of different subjects are compared to determine the difference in their identities.

In this work, the inception-based [18] FaceNet model is used due to the improved validation performance compared to the Zeiler&Fergus [19] model. Instead of focusing solely on reducing the difference between the embeddings of images of the same subject, an additional comparison is added to simultaneously increase the difference between the embeddings of different subjects: Triplet-Loss [16]. ArcFace employs a similar approach on a ResNet-based architecture, but compares the angular representation of the feature outputs to increase the distance between dissimilar subjects while increasing the stability of the training [17]. Both methods have been well established as state of the art facial recognition systems [12]. We utilize FaceNet as our primary FRS for evaluating our morphs while using ArcFace as a secondary evaluation model.

2.1.2 Attacking with Morphs

Although not the only threat morphs pose, fooling a facial recognition system is a key objective. Ferrara *et al.* in [11] presented a scenario in which a hostile subject submits an image for a passport that is a morph between themselves and another bona fide subject. The morphs generated for their study were made using manual techniques, and shown to be a viable threat to FRS. With the development of automated morphing techniques, the threat of morphs to FRS grows concerning as their

generation becomes widely accessible.

2.2 Morph Image Generation Techniques

2.2.1 Facial Landmark Morphing

There have been several automated face morphing techniques published in recent years, but most of which can be classified as either landmark-based or GAN-based. The latter utilizes the latent space of a generative adversarial network (GAN), whereas landmark-based techniques map the subjects' facial features and brings them closer toward each other [20, 21, 22, 1]. It has been documented that landmark-based morphing is superior to GAN-based methods [12, 13], so experiments were done using an open-source morphing technique utilizing landmarks to compare against the GAN generated morphs: Face Morpher [1].

Face Morpher [1], written by Quek, begins by calculating the landmarks of both subjects' face. Utilizing Dlib's detector, 68 landmarks in total are predicted surrounding the eyes, nose, mouth, chin, and the upper edges of the eyebrows [23]. These points are used to calculate a convex hull, cropping out a mask of each subject. The generated points are averaged to a set of points equidistant to each subjects' points, and using Delaunay Triangulation, the triangles of each subject are warped toward the averaged points. After warping, the faces are structurally aligned. Different blending approaches are used to combine the pixel values of the two faces to create an averaged mask. The warped and averaged masks are then pasted onto both original subjects; this results in two morph images of the bona fide subjects. Example

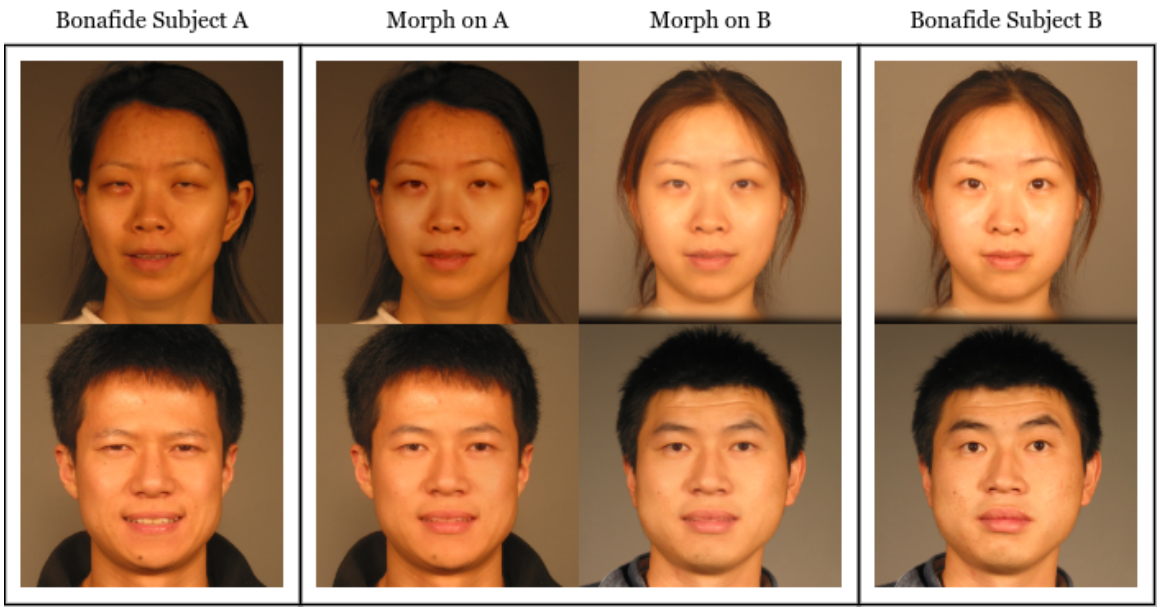

Figure 2.1: Example landmark-based morphs using [1] on the FRGC dataset.

morphs using FaceMorpher are shown in Figure 2.1.

2.2.2 Generative Adversarial Networks for Morphing

Generative adversarial networks have been a hot-button topic since their inception [5]. GANs involve two networks, a generator and a discriminator, pitted against each other during training. Both the generator and discriminator are traditionally mirror images of the other in both layer type and dimensionality (convolutional layers or fully connected layers for example). During training, the generator takes in a random noise input, generates some output, and the output is then fed to the discriminator. The discriminator then determines whether the output from the generator is a valid/real output or an invalid/fake output. This result serves as the loss to propagate through both networks. The discriminator is also trained on valid/real outputs using a separate loss function. Thus, the final result is a generator capable of generating

outputs similar to the real input images and a discriminator capable to distinguishing between the two.

The noise serving as the input to the generator is referred to as a latent code or representation. By using a latent representation of a particular output, we are able to change attributes of the output in the latent space. For example, a generator for making landscapes along with the latent code for generating a landscape of a tree-filled valley could be manipulated to change the weather, the color the trees, or add clouds in the sky. This opens up a new frontier for areas like graphic design [24], animation [25], and face editing [26].

One of the earliest GAN-based face morphing techniques used the MorGAN architecture [27]. Both the generator and discriminator are constructed using convolutional layers along with rectified linear unit activations (ReLU) for nonlinearization [28]. In this example, an encoder is also trained to estimate the latent representation of a given input image. The loss for this encoder takes into account a pixel-wise loss of the input and output image and an adversarial loss derived from the cross-entropy of the discriminator and generator losses. Thus, the encoder is trained along with the generator and discriminator. The latent code dimensionality for the MorGAN generator is 1×512 which is convolved into an output image of size $64\times64\times3$. To morph two subjects, linear interpolation of their corresponding latent codes is inputted into the generator to produce a morph of the two images.

In a study by Venkatesh *et al.* [12], landmark-based and GAN-based morphed images were evaluated against FRS. To evaluate the morphs' performance, they adapt a metric from [29] known as Mated Morph Presentation Match Rate (MMPMR).

MMPMR is calculated using a pre-trained FRS to produce a similarity (or dissimilarity) score between a given morph and images of the contributing subject. If the minimum similarity score from the comparisons is greater than a given threshold, the morph attack is successful resulting in a score of 1. The cumulative average of the scores produce the success rate of the morphs (see Equation 2.1. Landmark-based morphs had significantly higher MMPMRs than the GAN-based approachs. Using ArcFace [17] as the FRS, the landmark-based morphs [21] had an MMPMR of 95%, whereas none of the MorGAN morphs were successful (MMPMR of 0%). Venkatesh *et al.* [12], in addition to MorGAN, also evaluated morphs generated using a style-based GAN known as StyleGAN [2]. The StyleGAN morphs achieve an MMPMR of 39%, showing significant improvement from the MorGAN architecture while under performing compared to landmark-based techniques. Our work strives to improve upon the StyleGAN-based results by adapting techniques used in landmark-based techniques to improve the quality and performance of the morph images.

$$MMPMR(\tau) = 1/M * \sum_{m=1}^{M} [\min_{n=1,....N_m} S_m^n] > \tau \tag{2.1}$$

2.3 Style Based Image Generation

Karras *et al.* [2] proposed an improved GAN architecture (StyleGAN) to generate high resolution images of a much higher quality than previous GANs. In addition, the network's design makes it ideal for mixing styles of different images in the latent space. Although the resultant images were of a high caliber, there were noticeable artifacts in some samples. StyleGAN2, presented by Karras *et al.* [3], corrects these artifacts

while improving the quality even further. In this work, we utilize the StyleGAN2 architecture due to the improved image quality.

2.3.1 StyleGAN2 Architecture

The StyleGAN [2] model is constructed by a series of convolutional and upsampling layers organized into blocks based on the resultant resolution of the image outputted by the final convolutional layer of the block. This design is based on [30] to generate images at high resolutions through a progressive learning approach. When training, the generator and discriminator only have two blocks of convolutional layers, producing an 8x8x3 output. As the loss of the lower resolution layers converge, higher resolution layers are added and trained again. This repeats until reaching the final resolution of $1024 \times 1024 \times 3$. Using this methodology, the network's training stabilizes while reducing the amount of time it takes to train [30]. The other significant change to the traditional GAN architecture [5] is the input to the generator. In place of the latent code input to the first convolutional layer, a constant, learned input is used. The latent code is instead mapped through a series of fully connected layers to generate an intermediate latent code vector of size 1×512, which is then put through an affine transform for each convolutional layer before being mapped into the network. Due to this significant change, latent codes from different images can be inputted together to produce an image with mixed styles. Figure 2.2 shows the general architecture of the StyleGAN2 model. The figure shows the blocks for resolutions of 4×4 and 8×8.

The key difference between the StyleGAN [2] and StyleGAN2 [3] models is the manner in which latent codes influence the generator at each layer. In the StyleGAN

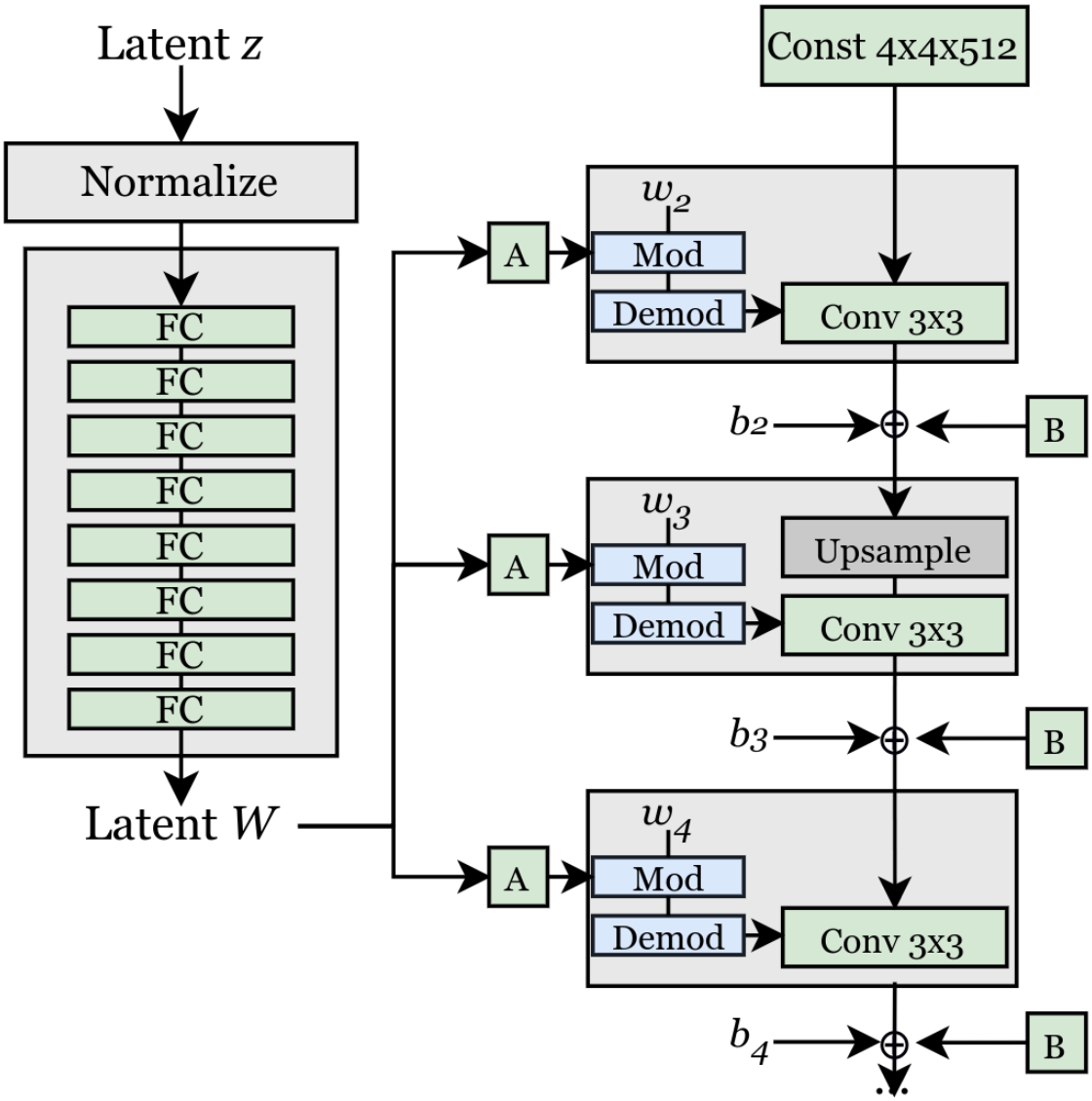

Figure 2.2: StyleGAN2 architecture generated from [2, 3].

architecture, a normalization method uses the output of the intermediate latent code and the affine transform to scale the normalization of each feature map. This adaptive instance normalization allows for the style to influence the feature maps at each resolution to manipulate the final output. For StyleGAN2, the artifacts produced by the normalization were corrected by replacing the normalization approach with a weight modulation and demodulation operation. This new approach scales the weights of a given convolutional layer using the inputted style vector and then scales down the product by its L_2 norm. The overall image quality improves slightly with this replacement in addition to the removal of the artifacts present in original output images.

2.3.2 Image Inversion

Morphing in the latent space requires a latent representation of both subjects, allowing the generator to reproduce the original images. The two general strategies for inverting from image to latent code include optimizing for each latent code separately [4] or training an encoder to convert an image into its latent representation [31, 32]. We only focused on the optimization latent embedding methodology in this work.

In [3], the authors discuss an optimization-based approach to invert or embed images into the StyleGAN2 latent space. A starting latent code is calculated from 10,000 random codes once mapped through the fully connected layers, resulting in an averaged latent code, W, where W is a 1×512 vector. To optimize the latent code, the perceptual loss [33] between the original image and the current synthesized image is back-propagated through the network to the latent code W. Learned Perceptual Image Patch Similarity (LPIPS) extracts features from both images to compare their

likeness:

$$L_{LPIPS} = \sum_{i=1}^{n} (E(x) - E(g(W+))^2, \tag{2.2}$$

where E is the LPIPS embedding representation for the down-sampled images, g is the StyleGAN2 generator, and n represents the size of the embedding. In addition, Gaussian noise is added to W for the first three-quarters of the optimization steps for increased stability and to traverse more of the latent space, assisting the finding of the global optimum [3]. The noise input to the feature maps throughout each layer of the generator is also learned to find the optimal noise compliment to latent code W. Training for noise can lead to the latent code containing less information of the style of the image with the noise containing significant information on how to reconstruct the image. To prevent this, noise regularization loss is added to restrict the noise to the form of a normal distribution:

$$L_{noise} = \sum_{i,j} L_{i,j} \tag{2.3}$$

where i is the layer of noise, j represents the amount of down-sampling performed on the given noise matrix. within the noise matrix, and $L_{i,j}$ is the regularization term for a given layer of noise. The total loss using this technique is the sum of both terms:

$$Loss_{Total} = \lambda_1 * L_{LPIPS} + \lambda_2 * L_{noise} \tag{2.4}$$

where $\lambda_1 = 1$ and $\lambda_2 = 10^5$. A standard Adam optimizer is applied over 1000 steps

to learn both the latent representation and the corresponding noise maps. Figure 2.3 shows projected examples using this technique. Example images were taken from a random input to the StyleGAN2 architecture (a) and from the Face Recognition Grand Challenge (FRGCv2) dataset [34]. We compare the target and reconstructed images using FaceNet as a verifier to find an averaged distance between the target and synthesized images. The average distance for the images in columns (a) and (b) is 0.36; the average distance for images in columns (c) and (d) is 0.70.

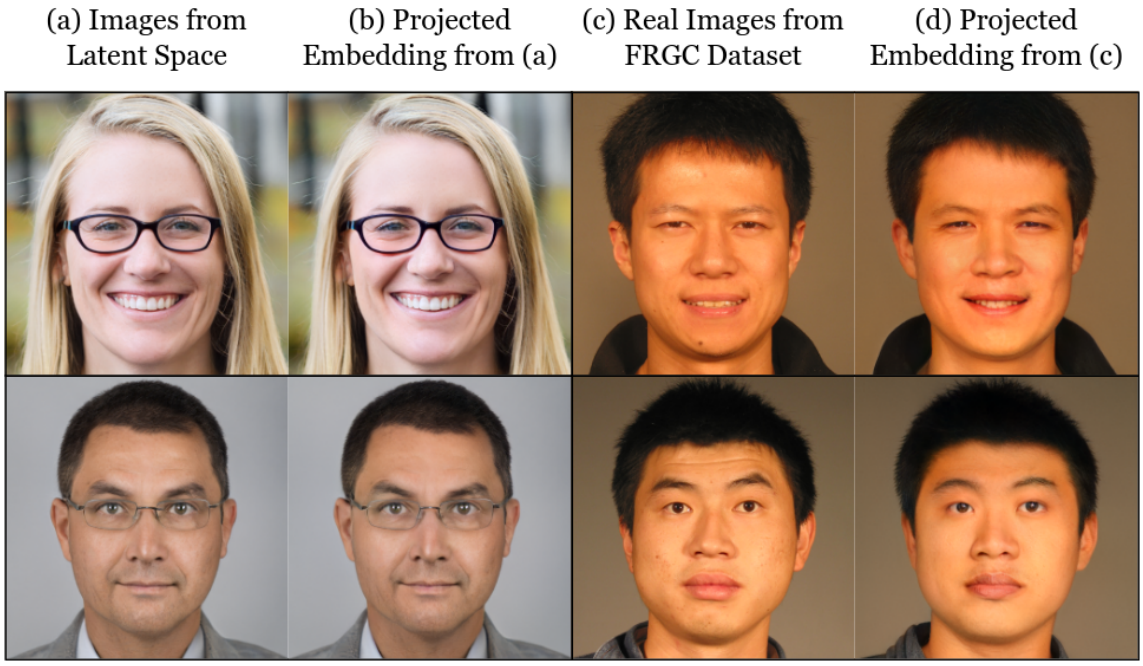

Figure 2.3: Example images embedded using the technique presented by [3].

The embedding methodology in [3] works well for images the generator has seen during training; however, the quality of the reconstructed images degrades when applied on new images. Here, the latent code is a 1×512 vector which is applied to each layer of the generator. This restricts the combination of styles that can be applied to a single image to a combination that exists in the latent space. Abdal *et al.* [4] expand the size of latent code to 18×512 to allow a different vector for

each layer. The latent code W is replaced with the extend latent code $W+$. A modified perceptual loss function now uses a pretrained Very Deep Convolutional (VGG) model to extract features from both the original and synthesized image [35]. The new perceptual loss between the target and synthesized image is calculated by comparing their features from four different layers of the VGG model. They compare the outputs of the $conv1_1$, $conv1_2$, $conv3_2$, and $conv4_2$ layers to derive their perceptual loss function:

$$L_{VGG} = \sum_{j=1}^{4} \frac{1}{N_j} ||(V_j(x) - V_j(g(W+))||_2^2, \tag{2.5}$$

where x is the target image, g is the generator, $W+$ is the latent code, V_j is the feature outputs of the VGG16 model for the layer specified by j, and N_j is the number of scalars in the feature maps for layer j. Pixel-wise loss is also added to measure the L_2 distance between the target and synthesized image:

$$L_{pixel} = \sum_{i=1}^{n} |x - g(W+)| \tag{2.6}$$

where x is the bona fide image. Structural and pixel level information is learned by the latent code by combining both losses:

$$Loss_{Total} = \lambda_3 * L_{VGG} + \lambda_4 * L_{pixel} \tag{2.7}$$

where $\lambda_3 = 1$ and $\lambda_4 = 1$. Only the latent code is optimized using this technique, so there is no noise regularization loss. The quality of the reconstructed images improve significantly for images the network has not seen during training (see Figure 2.4).

The average FaceNet distance for the reconstructed images in columns (b) and (d) are 0.315 and 0.262 respectively, showing a significant improvement in the inversion method compared to [3]. Although the latent space may contain the necessary styles to generate a new face, by expanding the dimensionality of the latent code, variability in the combination of applied styles increases. We explore both of these image inversion techniques to begin morphing in the latent space.

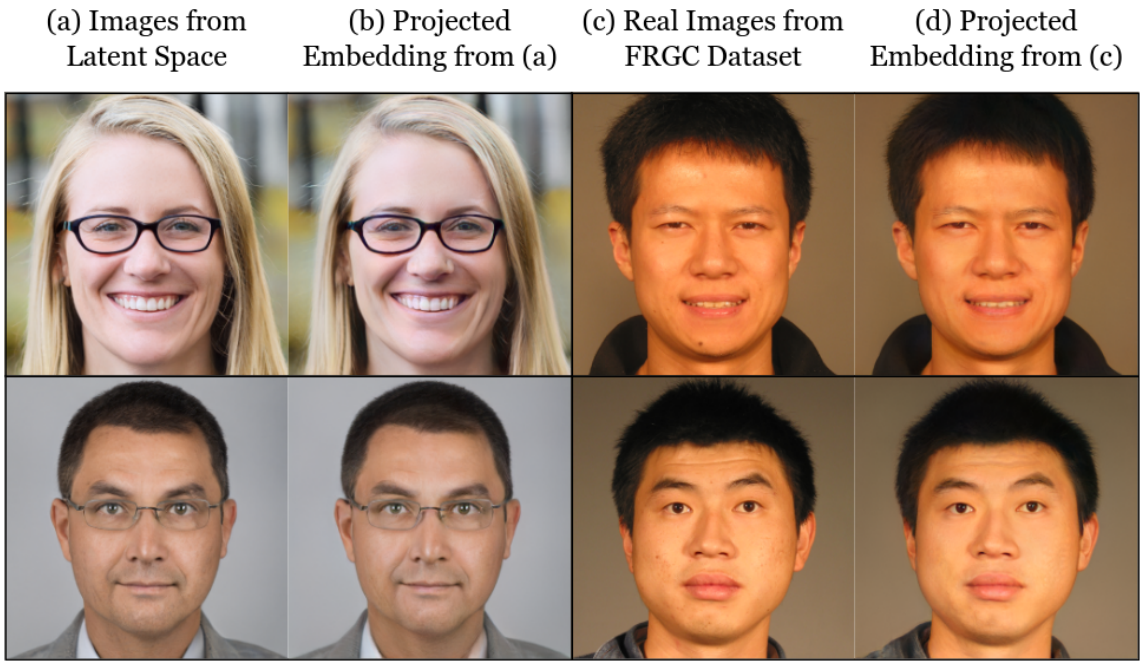

Figure 2.4: Example images embedded using the technique presented by [4].

Chapter 3

Morphing in StyleGAN2 Latent Space

3.1 Dataset Descriptions

For the following experiments, we use images from the Face Recognition Grand Challenge (FRGCv2) dataset [34]. FRGC contains images of subjects across different years and from two environments: hallway and plain backdrop. We develop pairs of identities for morphing from a subset used in [36]. In total, we use 165 bona fide subjects generating 305 morphing pairs of subjects with similar features. By using published pairings, we are able to compare our results to prior work by removing any difference caused by the pairings.

3.2 Standard Latent Embedding

We begin our experiments using published inversion techniques [3] [4]. Although the latter was originally designed for the original StyleGAN [2] architecture, the loss function and intermittent latent space work in a similar manner. Using both techniques, we morph the latent codes of two subjects simply by averaging the learned latent codes of the bona fide subjects.

3.2.1 Pre-processing and Inversion

StyleGAN2 is designed to produce images with equal height and width, therefore, the images used for the optimization steps are cropped down to $1024 \times 1024 \times 3$. In addition, each face must be centered within the cropped image, which is performed using [37] as recommended by the authors of StyleGAN [2]. Without the alignment step, features such as the eyes and mouth become corrupted when inverted as the network was trained on images that were centered using the same alignment method. Once aligned, we backpropagate for each image's latent code. Both techniques use an Adam optimizer [38] with beta values β_1 at 0.9 and β_2 at 0.999 which is applied over 1000 steps. In addition, we apply the same learning rate ramp-up and ramp-down method as [3] to stabilize training. The learning rate is ramped-up linearly over the first 50 steps from 0 to 0.1, and the ramp-down decreases the learning rate using a cosine schedule over the last 250 steps. The total time to embed an image using either technique is dependant on hardware configuration, so the average time varies between 150 to 450 seconds per image. After optimization, we save the final latent code (either a 1×512 or 18×512 matrix) and the final output image produced by the learned latent

code. Example reconstructed images using both techniques can be found in Figures 2.3 and 2.4. After learning, the morphs a generated by averaging the learned latent codes of a pair of bona fide images and inputted into the generator. After morphing, we evaluate the performance of the two sets of GAN generated morphs along with landmark-based morphs using [1] against FRS. Example morphs generated using [3] are shown in Figure 3.1; examples using [4] are shown in Figure 3.2

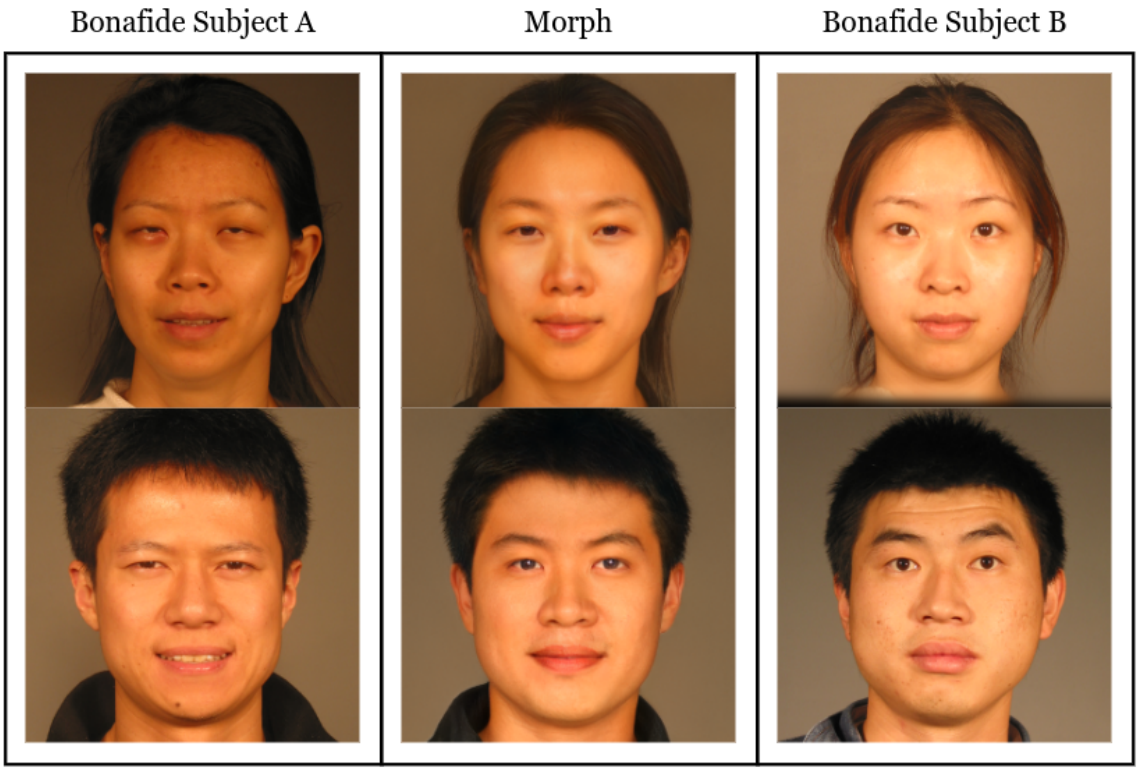

Figure 3.1: Example morphs generated by averaged latent codes learned using [3].

3.2.2 Morphing Results

Evaluation of the morphs using these inversion methods is performed using FaceNet [16] verification. We create a genuine pair using the bona fide images used in our

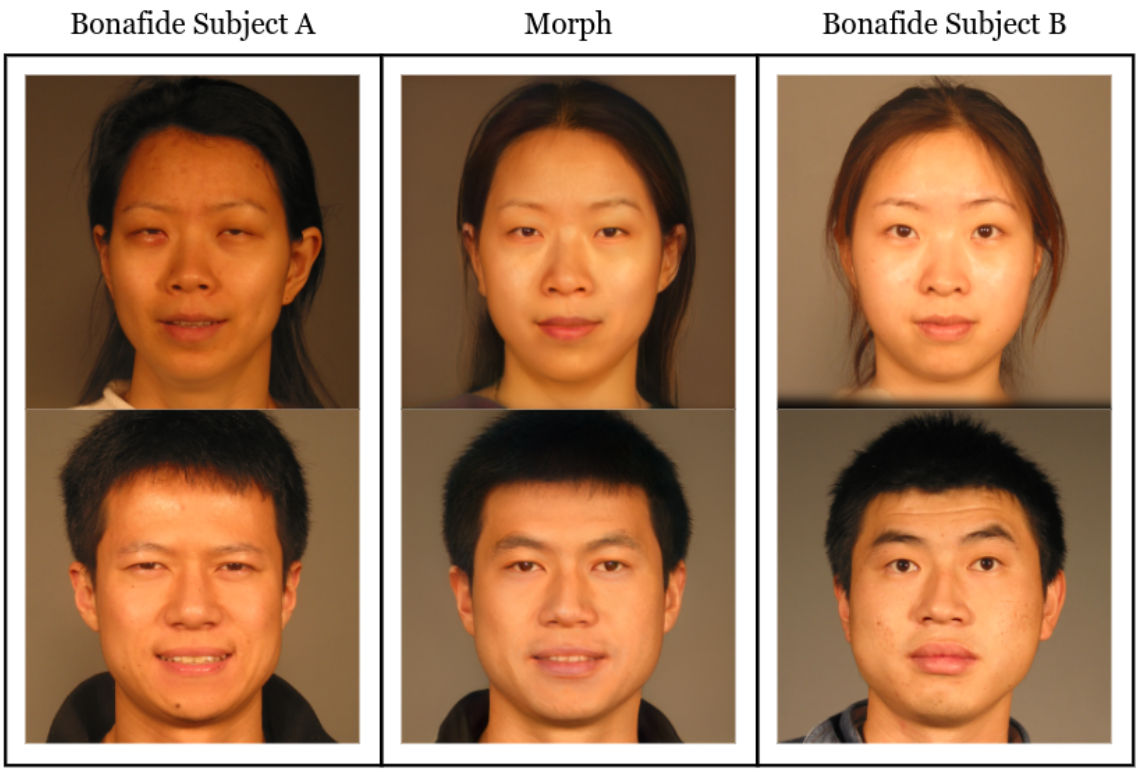

Figure 3.2: Example morphs generated by averaged latent codes learned using [4].

morphs and an alternate images of the same subjects. Imposter pairs are made using the bona fide images and their respective morphed images. Doing so, we compare every morph to both contributing bona fide images. The True Positive rate is the rate at which the genuine pairs are correctly classified as genuine, whereas the False Positive rate is the rate at which the imposter pairs are incorrectly classified as genuine. We plot the True Positive against the the False Positive to generate the Receiver Operating Characteristic (ROC) curve in Figure 3.3. In Table 3.1, we list the area under each ROC curve (AUCs) along with the Attack Presentation Classification Rates (APCERs). The APCERs show the percentage of morphs which fool the detector at select rates at which the verifier incorrectly classifies the genuine pairs (Bona fide Presentation Classification Error Rates or BPCERs). The APCER for the morphs

generated using [4] show an increase in the threat they pose comparatively to [3] morphs. For our purposes, we want the AUC to be closer to 0 and the APCERs closer to 100% showing an increased similarity between the morphs and the bona fide images.

Table 3.1: Morph Results Using Established Inversion Methods

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
FaceMorpher [1]	0.938	3.934	20.98	29.84
StyleGAN2 [3]	0.995	1.311	1.475	1.475
Image2StyleGAN [4]	0.983	1.485	3.135	4.455

3.2.3 Summary

Our morphing results using both inversion methodologies are comparable to other studies [12, 13]. The morphed faces are realistic, but there is a lack in identity as shown by their FRS verification results. Without passing verification, the morphs are not a threat to FRS. The modified Image2StyleGAN [4] image inversion method performed best, but without the learning rate ramp-up and ramp-down from [3], the inversion technique was unable to invert a significant percent of the dataset, resulting in a blank image. In terms of visual quality, the faces are realistic, but hair, clothing, and jewelry become distorted. Morph pairs where one has short hair and one long leads to "floating" hairs to form (see the morphs in the top row of Figure 3.2). We then explore two additions to improve identity preservation when morphing while removing the artifacts formed when morphing in the latent space.

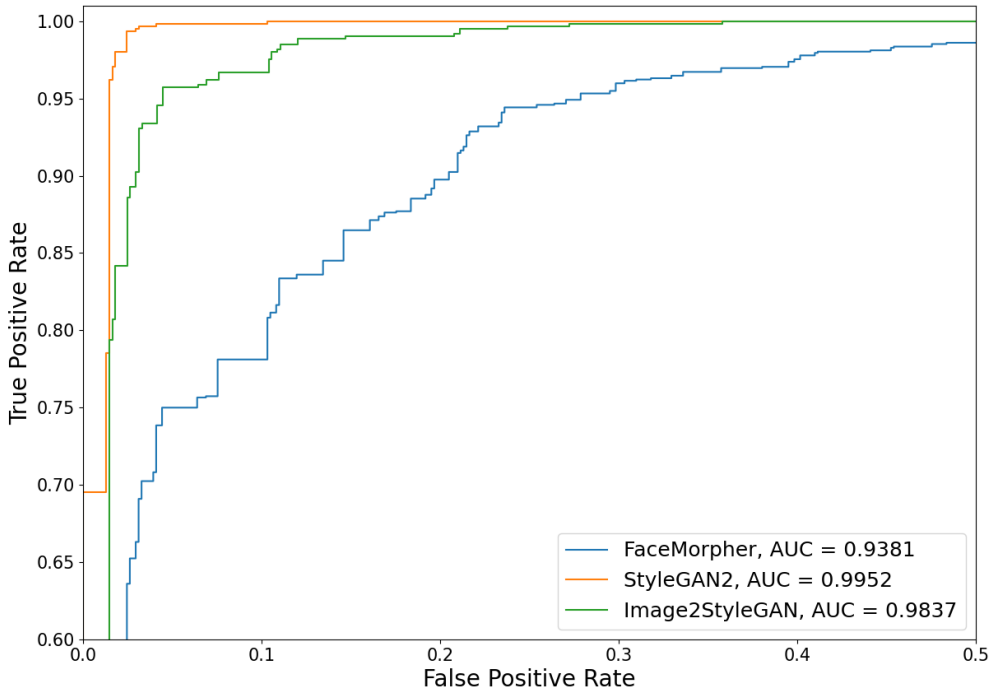

Figure 3.3: ROC curve comparing the FaceNet performance of StyleGAN2 morphing methods using [3, 4].

3.3 Latent Embedding of Convex Hulls

There are two key issues with the previous morphs: loss of identity and unrealistic border features, whereas landmark-morphing performs well in terms of identity preservation when morphing. We take the landmark warping step from landmark-morphing approaches to see the effect it has on the morphs generated in the latent space. Border features in our case include the hair, clothing, and jewelry present in the images. The hair is a finer detail of a face, which is produced by higher levels of the network influenced by both the latent code and the noise added to that layer. By averaging the latent codes of subjects with different higher level features, the morph

gains a mixture of features that a real image could not naturally posses. We create

3.3.1 Pre-processing and Inversion

We begin by performing the same aligning and cropping methodology from Section 3.2.1. We then warp the landmarks of the aligned images toward an averaged set of landmarks for each pair of subjects (see Figure 3.4). We adapt Quek's landmark-based morphing technique to estimate 68 landmarks for each subject, average them, and finally warp them using Delaunay Triangulation [1]. Warping the landmarks before inverting the images removes the morphed latent code's effect on the landmarks, as the latent codes now pose the same landmarks. As long as the inversion method preserves the landmarks of the warped images, the landmarks of the morphs will be comparable to landmark-based techniques.

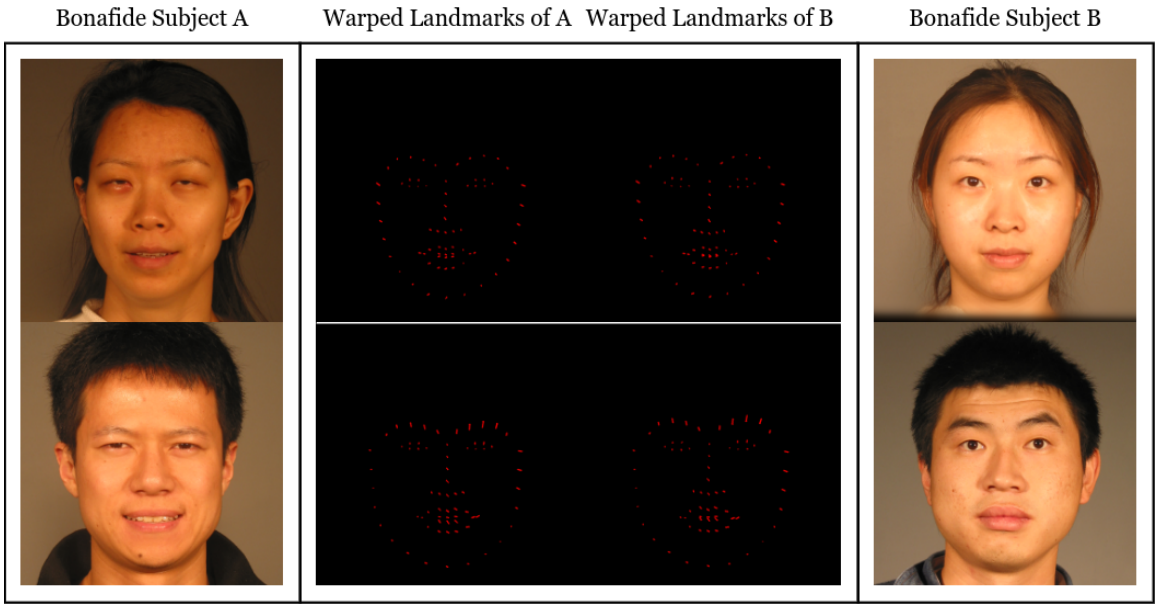

Figure 3.4: Example shows the shifted landmarks from the bona fide images to the average landmarks of the subject A and subject B.

The next change resolves the problem of the higher level features becoming distorted and unrealistic when morphing. We again modify the methods from [1] to isolate the face from the image, cropping out a mask as seen in Figure 3.5. In the original FaceMorpher methodology, the faces are cropped, warped, a convex hull is then removed from each warped image, they blend the pixel values, and finally paste the morphed convex hull onto the background of the bona fide subjects. By keeping the bona fide backgrounds, features such as hair, clothing, and jewelry would be preserved. Adapting this technique, we warp the landmarks and then save a mask of the warped faces of each subject. These masks or convex hulls will serve as the input to the latent code optimization method.

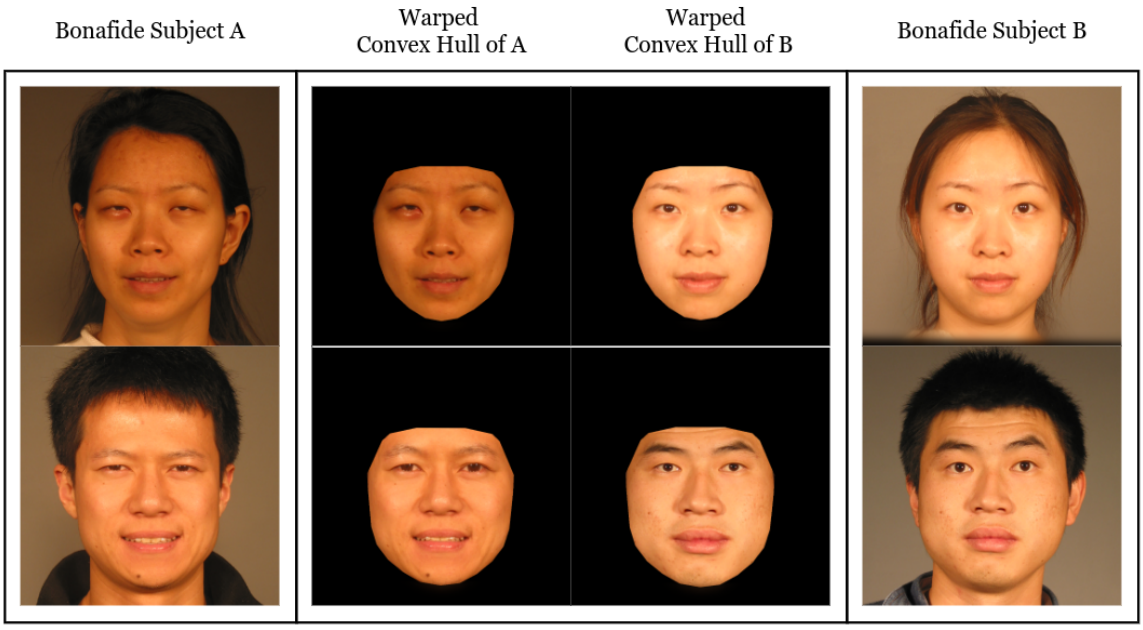

Figure 3.5: Example shows the warped convex hulls of the subject A and subject B.

The Image2StyleGAN method was superior in the quality of both the inverted and morphed images; however, we found inverting the convex hulls occasionally fell into a local minimum where the image becomes a solid black image. We found that the perceptual loss using layers from a pre-trained VGG16 model lead to the issue,

so we replaced their perceptual loss with [33], which was used in [3] (Equation 2.2). We also use pixel-wise loss (Equation 2.6) at a weight of 0.5 to avoid smoothing over the image. Noise regularization (Equation 2.3) is added, and finally a new loss term to regularize the latent code. Latent regularization was introduced in Robert Luxemburg's StyleGAN2Encoder [14]. To prevent the latent code of each layer from going beyond the scope of the latent space, ultimately effecting the morph-ability of two subjects' latent codes, an L_1 penalty is applied to the latent codes. We weight the latent magnitude regularization penalty by a factor of 10^{-1}, allowing for an accurate, but editable, latent representation to be found:

$$L_{reg} = \sqrt{1/N(W+)^2} \tag{3.1}$$

where N is the total number of latent values ($18 \times 512 = 9216$). Our total loss function for embedding convex hulls is:

$$Loss_{Total} = \lambda_1 * L_{LPIPS} + \lambda_2 * L_{noise} + \lambda_4 * L_{pixel} + \lambda_5 * L_{reg} \tag{3.2}$$

where $\lambda_1 = 1$, $\lambda_2 = 10^5$, $\lambda_4 = 0.5$, and $\lambda_5 = 0.1$. This loss function allows each convex hull to be inverted into its latent representation. Once we calculate the latent representations of a pair of warped convex hulls, we average the latent codes to generate the morphed convex hull of the two subjects. The convex hull is pasted onto the both bona fide subjects, following the same steps as FaceMorpher [1]. We display new morph examples in Figure 3.6.

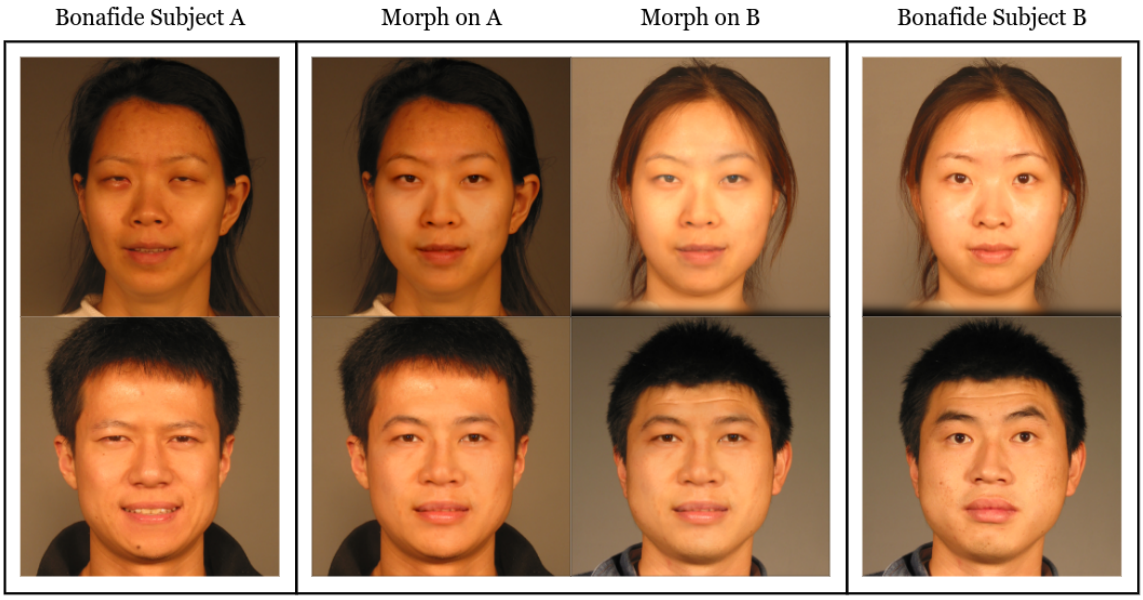

Figure 3.6: Example morphs generated by averaged latent codes learned using warped convex hulls.

3.3.2 Morphing Results

We again utilize FaceNet [16] to compare every morph image to each contributing bona fide image, plotting. Although the new morphs perform better than morphs generated by averaging the latent codes using [3], our AUC is higher than morphs using the latent codes from the Image2StyleGAN method [4] as shown in Table 3.2 and Figure 3.7, showing a greater difference in the similarity between the new morphs and the bona fide images. The drop in performance is due in part by the pasting step we perform after averaging the latent codes. The morph is pasted onto each of the contributing bona fide subjects and the colors are changed to seamlessly blend the mask with the background. This improves the quality of the morph compared to the bona fide subject whose background was applied; however, the performance compared to the other contributing bona fide subject degrades. When running verification, we compare each pasted morph to the contributing bona fide subjects. If we only compare

the morphs with the bona fide who's background is applied, the AUC decreases from 0.9878 to 0.9827. The APCER rates also improve with the APCER @ BPCER=5% increasing from 3.2787 to 7.5409.

Table 3.2: Morph Results Using Landmark Warping

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
StyleGAN2 [3]	0.9952	1.3114	1.4754	1.4754
Image2StyleGAN [4]	0.9836	1.4851	3.1353	4.4554
Warped StyleGAN2	0.9878	1.4754	2.4590	3.2787

3.3.3 Summary

By applying landmark warping before inverting the bona fide images into their latent representations, we are able to generate morph images with a comparable threat level than when using [4] to invert the images. Although the FaceNet verification performance degrades slightly, the resultant morphed images are less likely to be flagged by a human inspector due to the lack of artifacts in the hair, clothing, and jewelry in the images. The most significant change in our methodology to previous methods is the convex hull. StyleGAN2 was not trained to reconstruct a convex hull of a face but the entire head. Using the same methods for embedding a full head cannot be applied for the convex hulls. To overcome the inherit problem of the convex hulls, we explore the StyleGAN2 model to augment the embedding methodology to improve the latent representations for the convex hulls.

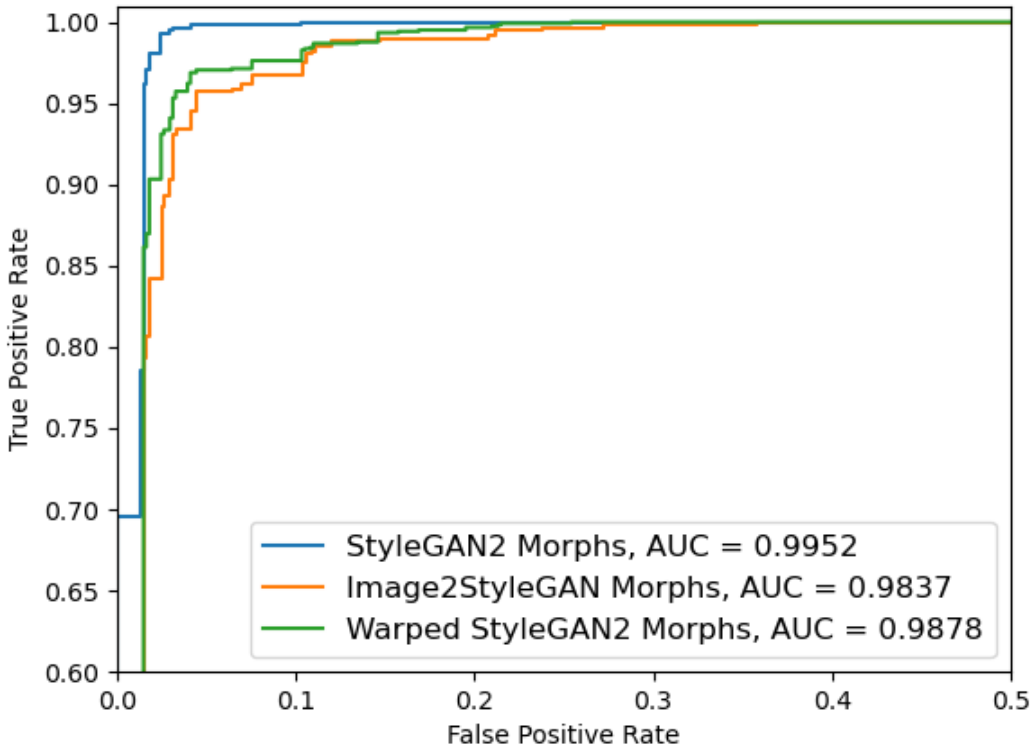

Figure 3.7: FaceNet ROC curve comparing standard StyleGAN morphing methods to our Warped Landmark Morphs.

3.4 Image Projection Experiments

Improving the latent representations of the convex hulls is the first step to improve the performance of GAN-based morphs using landmark warping. We first examine the influence of the noise input on the reconstructed images. We then explore methods for optimizing the noise as a means to improve the identity in the reconstructed images. Finally, the loss function for latent optimization is modified as a result of the noise experiments to improve the quality of the latent codes and the performance of the morphs. During exploration, we use the subset of images from the FRGC dataset used in Section 3.2 [34].

3.4.1 Train-ability of Noise

3.4.1.1 Training Noise before Morphing

Each layer of the StyleGAN2 architecture has a noise input, which adds finer details to the image at each resolution [3]. The noise is applied to the feature maps outputted by the convolutional layers, and the noise is generated from a random Gaussian distribution. By setting the noise to zero, we remove all texture from the faces ((a) in Figure 3.8). Adding noise with high values distorts the images by adding too much texture ((d) Figure 3.8). As a means to explore the significance of each layers' noise input, we remove the noise for all layers expect for one. Figure 3.9 shows a noise being applied to a single layer. Starting from the top left, Figure 3.9 shows noise being applied to the 4x4 resolution only and continues left to right for the next resolution. Noise in the early layers has little visual impact, whereas the higher layers

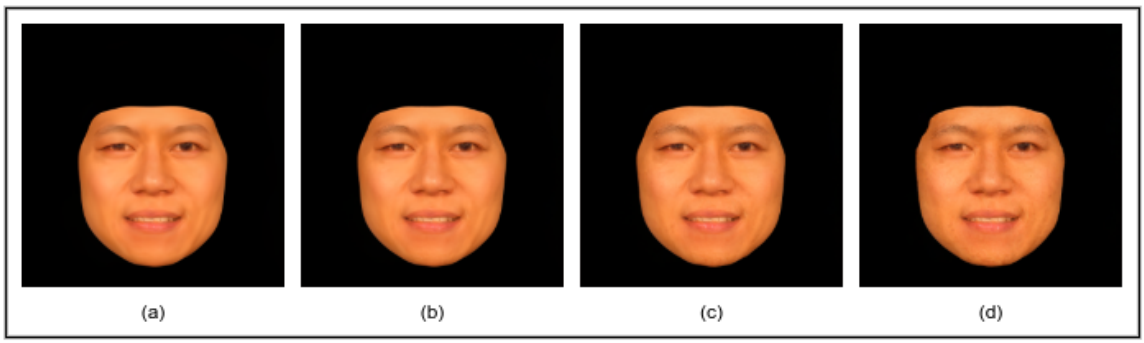

Figure 3.8: Example showing noise applied at increasing factors from 0 (a) to 1.5 (d).

(particularly the 128×128 layers) present a significant change in the texture. From this, we can say the noise is essential in the reconstruction of an image, but the question of how best to apply it needs addressed.

We then explore the train-ability of the noise inputs for a particular image. After training for the latent representation, we freeze the values of the latent code and train for a list of noise inputs. Perceptual loss is removed, so we only have pixel-wise loss being applied as the latent representation is responsible for the structure of the image. The learning rate is ramp-ed up similar to [3] and ramp-ed down starting at step 400 as finding the noise requires smaller adjustments earlier in training than with the latent codes. To reduce the time to optimize the noise, we add a verification check based on FaceNet to stop the optimization if the L2 distance between the bona fide image's and the reconstructed image's embeddings fall below 0.04. After training for the noise, we take the final reconstructed images and paste them onto the warped bona fide images. We evaluate the performance of the new inverted images to the images generated using [3, 4] to compare the inversion methodologies. We then explore blending the noise values in addition to the latent codes by averaging the noise values of each subject and applying it when reconstructing the average latent codes.

Figure 3.9: Example showing noise applied to each resolution block of the network separately.

3.4.1.2 Results

We first compare the performance of the inverted images we generate in Sections 3.2 and 3.3 using FaceNet [16] verification and the same genuine pairs previously used (see Table 3.4 and Figure 3.10 for complete results). An inversion method's performance is effective when the AUC is below 0.5 meaning the inverted images before similarly to the alternative images used in the genuine pair. The inversion method from [3] preforms poorly with an AUC of 0.99, whereas the Image2StyleGAN [4] method produces images with results better than the inversion of the warped convex hulls. However, the convex hulls come from the warped bona fide images after the first step toward morphing (as their landmarks have been shifted toward the average landmarks of two bona fide subjects). Therefore, the performance of the warped images serves as the maximum potential for the inversion methods using the warped convex hulls. As the Image2StyleGAN images' performance is far from the alternate bona fide images, we see the inversion method is not perfect. Although they are from warped convex hulls, the inverted images generated in Section 3.3 (named Inverted Warped Images) do not perform as well as the warped images, showing a loss in the identity during the embedding process.

The trained noise, however, significantly improves the performance of the inverted convex hulls. Various aspects of the images improve including texture, detail around the eyes, hair, and mouth. Figure 3.11 shows the improvement in the reconstructed images when we shift from no noise (center left) to random noise (center right) and the trained noise (right). In addition, due to the improved detail around the edges of the face, the artifacts produced by pasting the reconstructed mask onto the background of the bona fide image have been corrected as the images are near identical.

The performance of these images show results near identical to the warped images, removing any loss in the image quality after embedding and the optimization of the noise.

However, the noise values, when blended, distort the morph, and do not represent a blended texture of the bona fide subjects. This conclusion is similar to an observation made in [3] when the noise regularization loss is removed when training for noise. By training for noise using pixel-wise loss only, the noise may be learning coarser details and not just finer details such as texture. In addition, each trained set of noises is only usable with the latent code inputted during its optimization. We conclude that training for the optimal noise value before morphing is not a viable option for latent-based morphing.

Table 3.3: Inversion Results on FaceNet

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
StyleGAN2 [3]	0.9900	1.2121	2.4242	4.8484
Image2StyleGAN [4]	0.6721	44.1718	76.0736	87.1165
Warped Images	0.7341	35.7377	65.2459	83.9344
Inverted Warped Images	0.9115	10.3279	21.4754	39.8361
Inverted Warped Images Trained Noise	0.7213	35.7377	69.8360	87.2131

3.4.1.3 Training Noise after Morphing

Although training for the optimal noise values for a given latent code does not improve the quality of the morph, noise values have a significant impact on StyleGAN2 generated images' verification performance. Instead of training for the optimal noise values for each subjects' latent codes, we train for the optimal noise values after averaging the latent codes. We blend the pixels of the bona fide images and use the

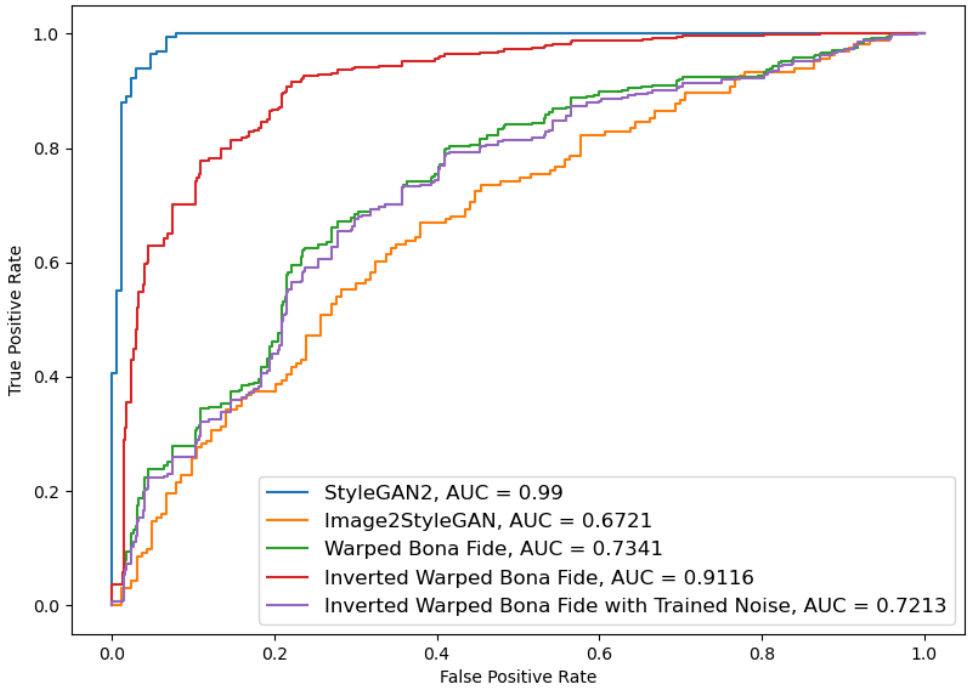

Figure 3.10: ROC curve showing the quality of the image inversion methods.

blended face to serve as the comparison for the pixel-wise loss, allowing for the optimal noise values for the morph to be learned. We keep the same hyper-parameters as the experiment in the previous section.

We note that training for the averaged pixel values combined with the landmark warping would result in training for noise values to make the morph appear near identical with the landmark-based approaches. To avoid this, a mask is generated using the landmarks surrounding the eyes, nose, and mouth and applied to the average of the bona fide images, removing those features from the average of the two bona fide images. The inverse of the mask is then applied to the reconstructed image from the averaged latent codes before noise optimization. By doing so, we keep the artifact

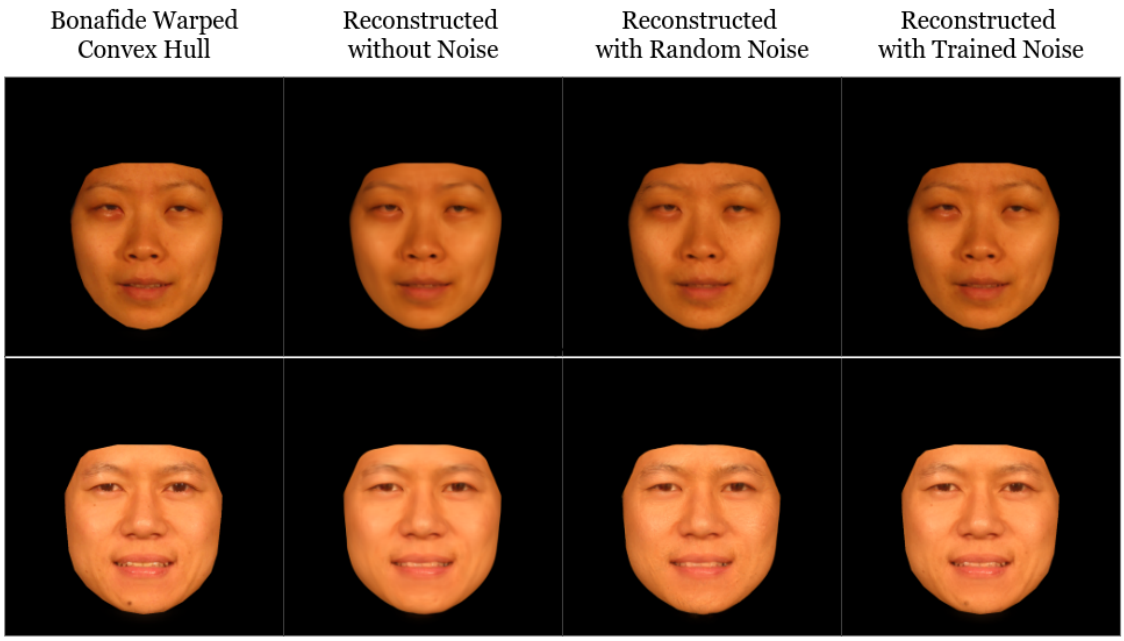

Figure 3.11: Example shows the warped convex hulls (left) after being embedded and reconstructed with different noise values being applied.

free eyes, nose, and mouth from the latent-based morph while training for the optimal morphed texture.

3.4.1.4 Results

Morphs using the trained textures perform better than the random noise-based approach. Visually, however, have more artifacts than when random noise was applied. In addition to the artifacts caused by pasting, boundaries around the eyes, nose, and mouth are pronounced in some examples. Further testing in learning texture may lead to an improved latent-based morph generation technique; however, we leave this problem for another project to try to different approach to improve the embeddings.

3.4.2 Noise Regularization Revisited

3.4.2.1 Experiment

Training for the optimal noise values for the latent code of a bona fide image improved the quality of the reconstructed image. However, when used with a different latent code, the noise distorted the output. We concluded the cause to be image specific information being added to the noise. Although noise regularization is applied during optimization of the latent codes, the applied noise still contains information to reconstruct the bona fide image. Latent-based morphing relies on the quality and amount of information represented by the latent codes. To remove any information from being stored in the noise instead of the latent code, we modify the latent code optimization technique by setting the noise to zero after the first 400 steps of the optimization.

Removing the noise too early causes the optimization to generate a corrupted image, but when removed after the first 400 steps, the latent code has learned enough information to prevent the corruption. The latent code then continues to optimize for the remaining 600 steps. Although the latent code cannot learn texture, by removing the noise, we aim to gain any information that was being capture by the noise instead to improve the quality of the latent representation of the bona fide images. In addition to the change in the noise values, we also reduce the pixel-wise loss weight to 0.05. This results in a total loss function:

$$Loss_{Total} = \lambda_1 * L_{LPIPS} + \lambda_2 * L_{noise} + \lambda_4 * L_{pixel} + \lambda_5 * L_{reg} \tag{3.3}$$

Figure 3.12: Results from the Image2StyleGAN [4] and the proposed method.

where $\lambda_1 = 1$, $\lambda_2 = 10^5$, $\lambda_4 = 0.05$, and $\lambda_5 = 0.1$. Figure 3.12 compares the contributing subjects (a) and (e) to the morphs generated using [4] and our proposed method. We show the morphed mask pasted onto contributing subject (a) in column (b) and the mask pasted onto contributing subject (e) in column (d).

3.4.2.2 Results

We see in Figure 3.13 a plot for the FaceNet [16] scores using L_2 distance for the warped images (a), inverted images from Section 3.3 (b), the new inverted images with the removal of noise (c), and the inverted images after noise optimization from Section 3.4.2 (d). The further to the left each distribution falls represents a greater similarity between the bona fide images and the imposter images. From these plots, we can see that the new embedding methodology does improve the latent representation's ability to reconstruct the warped bona fide subjects compared to our original performance. We repeat the same verification test from Section 3.4.2 to compare the performance of our updated inverted images against their bona fide image (Figure 3.14). The

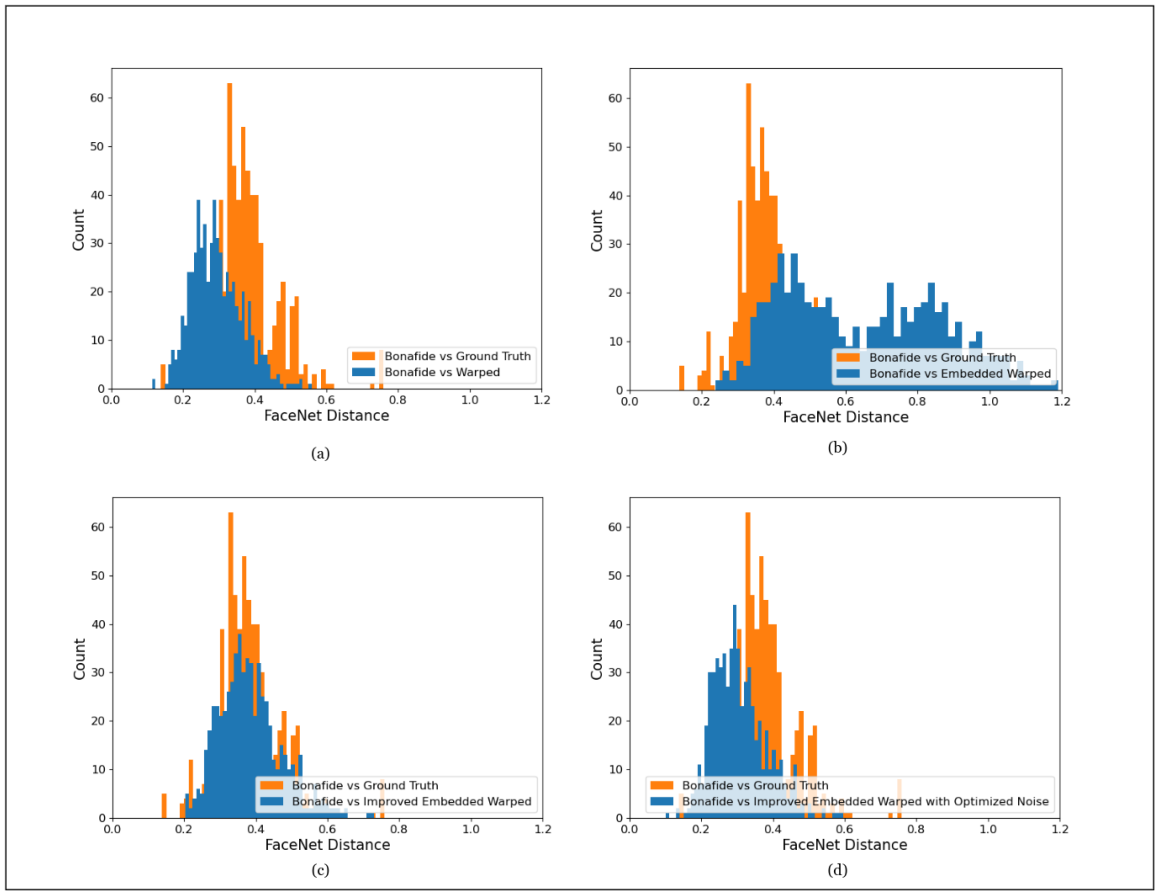

Figure 3.13: Histograms of FaceNet scores to evaluate the loss function described in Section 3.4.2

performance improves upon Section 3.3 with a decrease in AUC of around 0.1 and doubling APCER @ BPCER of 30% and 10% (Table 3.4).

Without noise, the reconstructed images do not posses any texture or finer details; however, regardless of the lack of texture, their performance on FaceNet [16] improves. We then take the new latent representations, average them, apply a random noise value as done previously, and compare the performance of the morphs (see Table 3.5 and Figure 3.15). The performance of both the average of latent code morphs using the improved embeddings for the warped convex hulls improves the AUC from 0.987 to 0.981 and improves APCER @ BPCER=5% from 3.27 to 7.54, showing a significant

Table 3.4: Updated Inversion Results on FaceNet

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
Warped Images	0.7341	35.7377	65.2459	83.9344
Inverted Warped Images	0.9115	10.3279	21.4754	39.8361
Inverted Warped Images Updated	0.8327	19.6721	41.1475	62.7868
Inverted Warped Images Trained Noise	0.7213	35.7377	69.8360	87.2131

improvement int he similarity between morphs and their contributing subjects. The morphs with the averaged noise pose a more significant threat to FaceNet, but due to the increased number of potential artifacts caused by the mask, reduces their threat against human inspectors.

Table 3.5: Morph Results using Landmark Warping with Improved Inversion

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
StyleGAN2 [3]	0.9952	1.3114	1.4754	1.4754
Image2StyleGAN [4]	0.9836	1.4851	3.1353	4.4554
Warped StyleGAN2	0.9814	1.4754	3.1147	7.5409
Warped StyleGAN2 with Averaged Noise	0.9659	2.4590	10.3607	18.3607

3.4.3 Summary

The combination of both perceptual and pixel-wise losses results in an improved latent representation of the bona fide images [4]. We also see an improvement in GAN-based morphs when the landmarks of the bona fide subjects are warped and cropped before being inverting the latent space. Without a good latent representation, the resultant morphs cannot fool FRS or a human inspector with any success. The performance, however, compared to the baseline inversion and averaging methods is minimal. Although the loss caused by pasting problem is still apparent, we recognize the biggest

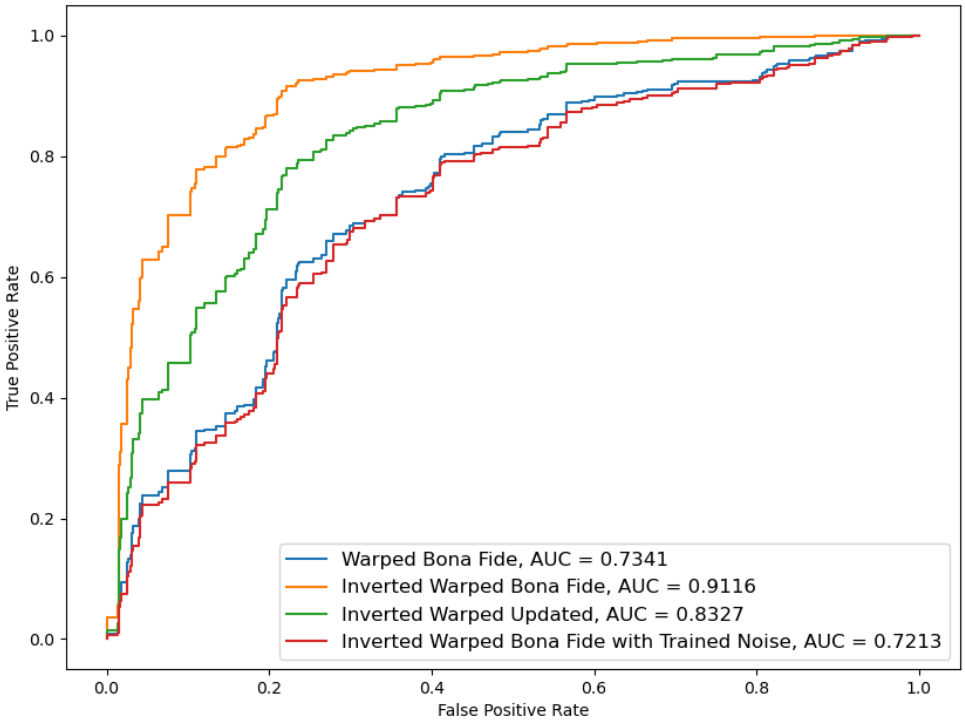

Figure 3.14: ROC curve showing the quality of the warped, inverted images.

drop in performance after warping occurs when the latent codes are averaged (see Tables 3.4 and 3.5). Our next experiment explores alternatives to averaging to further improve our GAN-based morphing results.

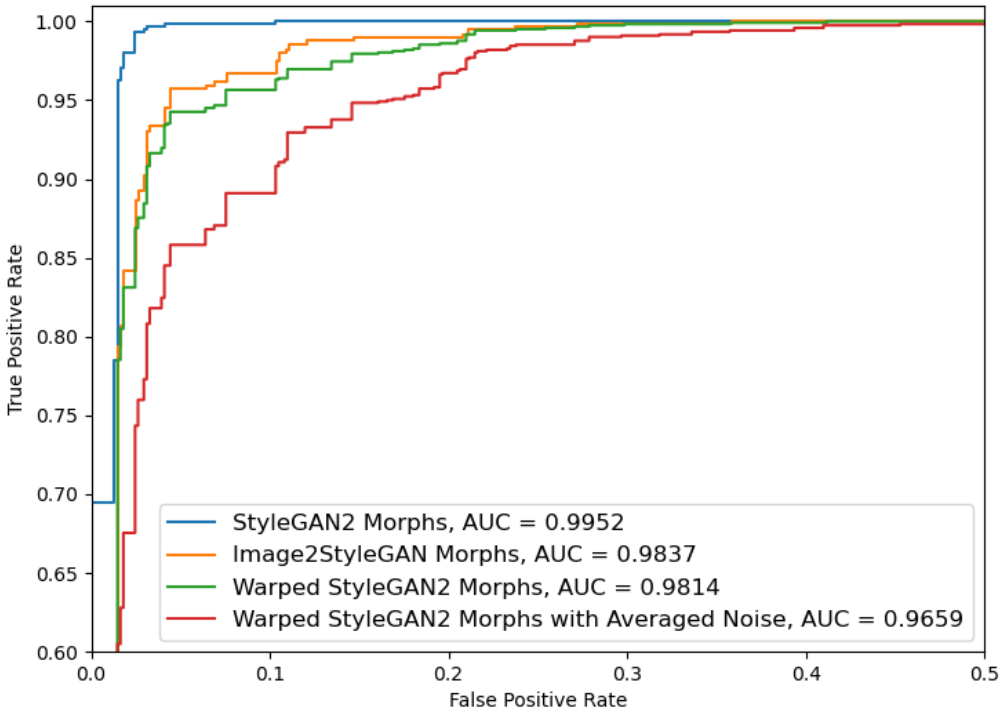

Figure 3.15: ROC curve comparing improved performance of landmark warped morphs and morphs with optimized noise values.

Chapter 4

Morphing through Latent Space Manipulation

Averaging latent codes assumes the latent space can be traversed linearly. After evaluation of the verification scores discussed in Chapter 3, the morphs generated from the average latent codes of two subjects are bias toward a single subject. Quality morphs must balance the identity of both subjects. We explore alternatives to averaging using Principal Component Analysis (PCA). After morphing, we finalize our morphing results using FaceNet verification, single morph detection, and MMPMR [29].

4.1 Datsets

For the following experiments, we again use images from the Face Recognition Grand Challenge (FRGCv2) dataset [34] used in prior experiments. We increase the number of bona fide subjects to 374 to generate 747 morphing pairs. The increased subset size does impact the final results shown in this chapter compared to the results shown in Chapter 3.

4.2 Latent-based Morphing via Principal Component Analysis

We first explore the effect of PCA when applied on the latent codes of the convex hulls from our previous experiments discussed in chapter 3. The latent codes have dimensions 18×512, so the of the data would be covariance matrix would be 9216×9216. However, we assume each latent code vector represents an independent style of the image, so we perform PCA on each layer/vector of the latent code separately. We will then have 18 covariance matrices of size 512×512.

4.2.1 Exploring Variance

The PCA models are first trained using a large dataset of latent codes from warped convex hulls generated from a dataset of twin images. We utilize the SciPy [39] library to train our model and generate our eigenvectors. Before projecting another dataset onto the model, we first explore the explained variance of the eigenvectors for each

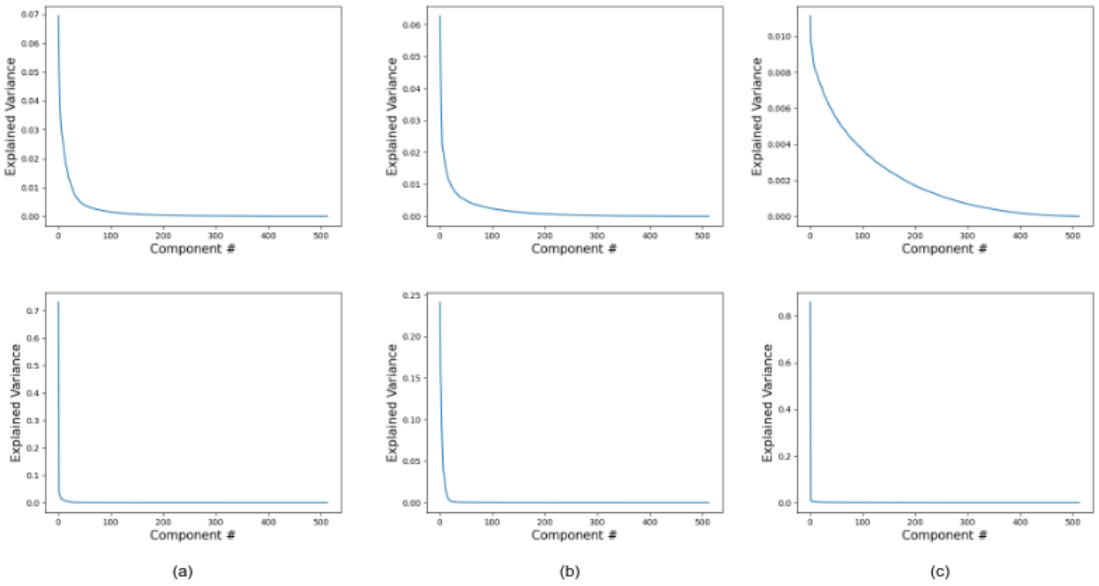

Figure 4.1: Explained variance of the principal component from three latent vectors (top row) and three style vectors (bottom row).

layer of the latent codes. The top row in Figure 4.1 shows the amount of variance the eigenvectors for three different latent code vectors. The values do not converge quickly; nearly 90% of the eigenvectors are required to reconstruct the original latent codes without significant loss of information. These eigenvectors are not ideal for the purposes of there is very little correlation between the latent codes. We go back to the StyleGAN2 network to find an alternative space in which to perform PCA [3].

Latent codes are not directly inputted into the convolutional layers of the StyleGAN2 [3] model. Before weight modulation, a learned affine transform converts the latent vector into the true style vector that will influence the weights of the convolutional layer. This linear transformation changes the values and the dimensionality of the latent code to match the dimensionality of the current layer. The style input for the resolution blocks up to the 64×64 resolution are 1×512 and are reduced by half

for each remaining resolution. Unlike the latent codes of which we have 18, there are a total of 26 style vectors. This is due to the layers of the network used to convert the feature maps into an image. The latent vector applied to the previous layer is put through the affine transform of this conversion layer, which generates another style vector we must morph. We convert all of the latent representations we've previously generated and repeat the steps to train our PCA models. The styles converge rapidly as shown in the bottom row of Figure 4.1. With the improved eigenvectors, we explore ways to morph the new style representations by only averaging projections on the first eigenvectors and varying the blending of the remaining projections.

4.2.2 Maxing vs L2 Norm

The first morphing technique is influenced by [40]. Wavelet decomposition breaks two bona fide images into a series of sub-bands representing different frequency content. The sub-bands containing the lowest frequency content of two bona fide subjects are fused by averaging, while the remaining sub-bands fuse by selecting the maximum values from the two. We apply the same methodology but on the reconstructed styles after projecting them back onto the eigenvectors. First, we select the amount of eigenvectors to use for averaging. The total number of eigenvectors is dependent on the current style vector we are projecting, so in place of a fixed number, we use a percentage. We evaluate the results for averaging thresholds of 60%, 50%, 30%, 20%, and 10% while using the remaining percentage using an alternative blending method. For each style vector, we project them into our pre-trained PCA space, calculate the number of eigenvectors using the current threshold, projects the styles onto the first eigenvectors up to the calculated amount, and projects them again on the remaining

eigenvectors. We then average the first projection results for a given morph pair. For the projections from the remaining eigenvectors is blended using either maxing or L2 norm selection.

For maxing, we go element by element through the remaining projected values of a given pair to generate a new vector containing the values with the greatest magnitude of the projected style vectors. By doing so, we select the information with the greatest value between the two styles, which we assume is the more significant information for reconstruction. The averaged and maxed vectors are added together, making the new morphed style for the given pair. We repeat this process for each style vector. The morphed style is inputted into the network to reconstruct the morphed image followed by the pasting step to blend the morphed mask onto each bona fide subject [1]. Figure 4.2 shows the bona fide images (a) and our Warped StyleGAN2 morphs (b) against morphs generated using element-wise maxing.

Our second approach uses a vector-wise selection technique as opposed to the element-wise selection when maxing. After computing the projection of the remaining eigenvectors, we compute the L_2 norm of the projected vectors. The norms are compared, and whichever has the greatest value, we select the entirety of that style vector's projection to combine with the averaged projection vector. This is done under the sumption that the style vector farthest away from the origin is more significant than the other. This approach, however, can result in bias within the morph toward one subject if their latent codes has a larger L_2 norm for the majority of their style vectors. We show example morphs using norm selection in Figure 4.3.

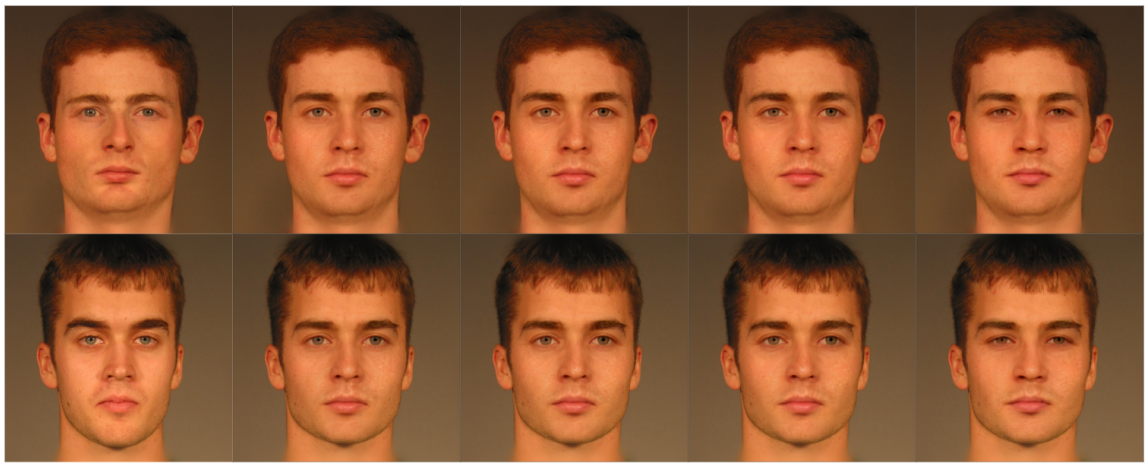

Figure 4.2: Example morphs using PCA and element-wise maxing. Compares the bona fide images (a) and Warped StyleGAN2 morphs (b) to the PCA morphs using thresholds of 60%/40% (c), 40%/60% (d), and 10%/90% (e).

4.2.3 Results

We first evaluate the new morphs using the same FaceNet verification method used previously to compare against our standard averaging method. To compare against alternative methods, we use morphs generated using a landmark-based approach [1], the StyleGAN2 approach [3], the Image2StyleGAN approach [4], and morphs generated using MIPGAN [36]. Our genuine pairs consist of the bona fide images used to morph and alternative images of the bona fide subjects, while the imposter pairs are the bona fide images and the morphs. Again, lower AUCs and higher APCERs corespond to a greater threat posed by the evaluated morphs. We list the AUCs and the APCER values at BPCER rates of 30%, 10%, and 5% in Table 4.1 where Warped StyleGAN2 are the morphs using our improved method and latent averaging to morph.

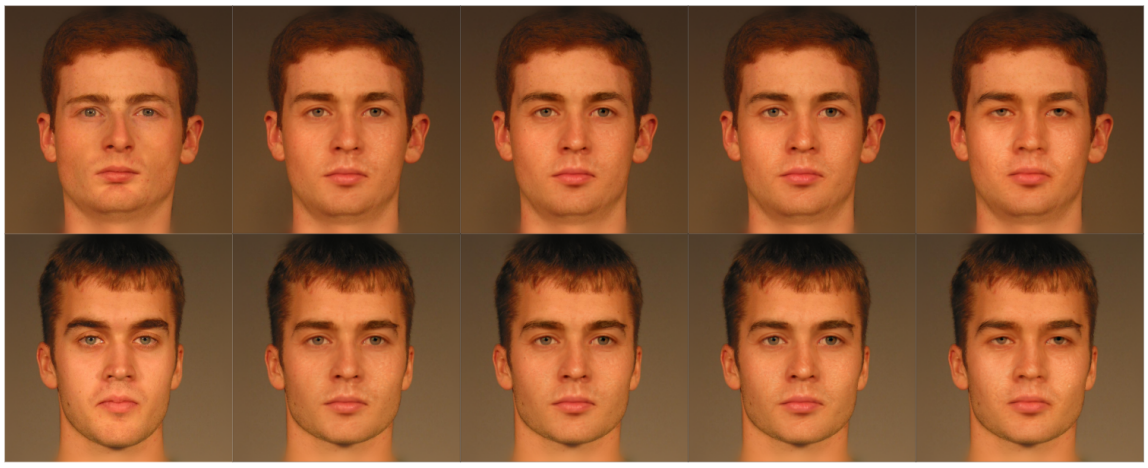

Figure 4.3: Example morphs using PCA and norm selection. Compares the bona fide images (a) and Warped StyleGAN2 morphs (b) to the PCA morphs using thresholds of 60%/40% (c), 40%/60% (d), and 10%/90% (e).

Table 4.1: Morph Results using Landmark Warping with Improved Inversion

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
FaceMorpher [1]	0.893	12.05	33.00	48.06
StyleGAN2 [3]	0.995	0.0669	0.6024	2.744
Image2StyleGAN [4]	0.967	1.218	11.77	18.94
MIPGAN [36]	0.979	0.8032	6.292	13.12
Warped StyleGAN2	0.955	2.744	16.00	24.63

Our Warped StyleGAN2 verification performance is superior to the other GAN-based approaches; however, the results fall short of the landmark-based morphs. One important note is the performance of the MIPGAN morphs [36]. The results shown in their work are not reflected in our testing results. We note that the images we used were generated using a distribution of their program, leading to potential misuse of their software. This difference in performance is addressed in [13] which explores additional explanations for performance difference. After establishing a baseline for the

above methods on FaceNet, we perform the same test on the morphs using element-wise maxing and norm selection with PCA. The AUC and APCER values are listed in Table 4.2, where Maxing 60/40 identifies the method in which the first 60% of eigenvectors were used for averaging and the remaining 40% were used in the element-wise maxing. We also plot the ROCs for the element-wise maxing in Figure 4.4 and norm selection in Figure 4.5, showing the performance of the morphs compared to both contributing bona fide images.

Table 4.2: FaceNet Performance on PCA Morphs

Method	AUC	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
Maxing 60/40	0.955	2.744	16.00	24.43
Maxing 50/50	0.957	2.744	16.00	23.76
Maxing 40/30	0.957	2.343	15.66	23.76
Maxing 30/70	0.961	1.205	14.12	22.62
Maxing 20/80	0.966	0.803	11.78	21.29
Maxing 10/90	0.973	0.803	10.04	17.80
Norm 60/40	0.954	2.744	16.93	26.24
Norm 50/50	0.957	2.343	16.00	24.43
Norm 40/60	0.956	2.343	16.73	24.97
Norm 30/70	0.958	1.205	16.73	24.97
Norm 20/80	0.953	1.205	17.80	28.98
Norm 10/90	0.942	2.343	22.22	37.22

Comparing the results from the PCA morphs and the averaged morphs, we see no improvement with the element-wise maxing approach. One explanation is element-wise maxing mixes the values in a style vector into a vector which may or may not exist naturally in the latent space. The norm selection method does improve upon the performance of our original morphing technique. The 60%/40% combination improves the APCER @ BPCER=5% by 2% showing only a slight improvement in performance, but the 10%/90% combination improves the APCER @ BPCER=5% by 12% and the AUC by 0.003. The FaceNet [16] performance improves by a small

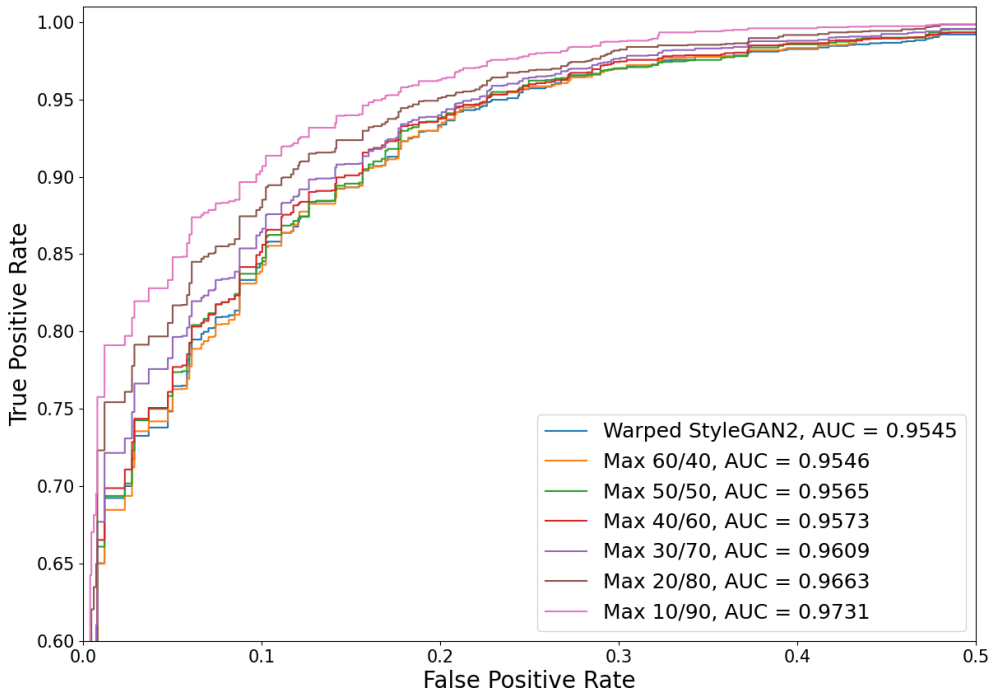

Figure 4.4: ROC Curve for morphs generated using PCA with element-wise maxing.

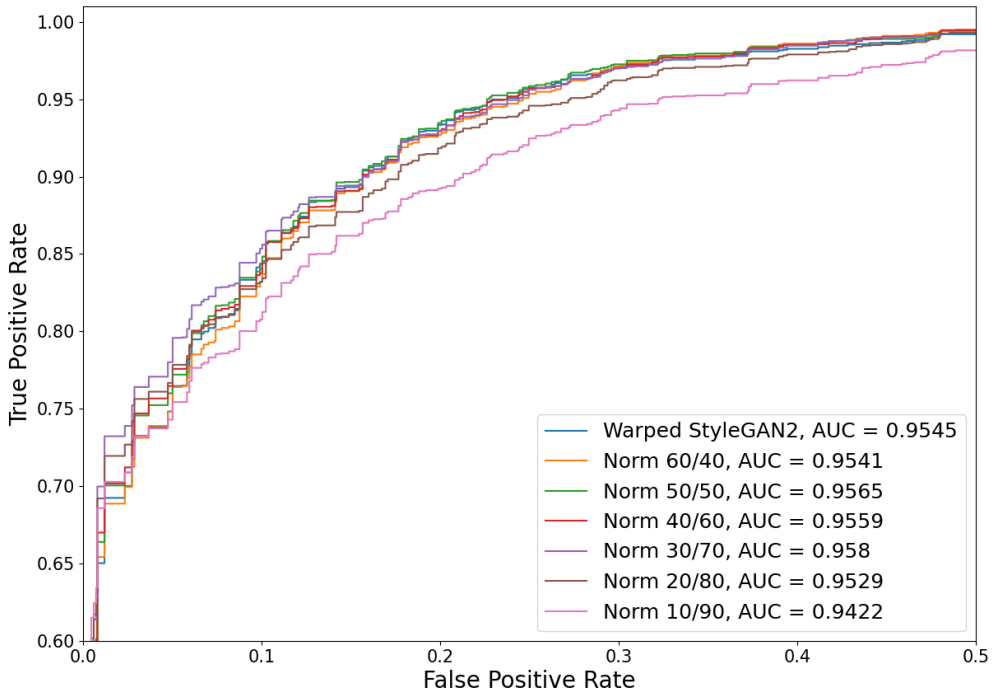

Figure 4.5: ROC Curve for morphs generated using PCA with norm selection.

amount, but this is using one evaluation method. Plotting the ROC curve only shows the performance of the morphs relative to the performance of the alternative images. To further analyze the performance of our morphs, we calculate the MMPMRs at a False Acceptance Rate of 0.1% as set by [41] using FaceNet as the target FRS. We first examine the performance of our Warped StyleGAN2 approach and the morphs generated using MIPGAN [36]. Results are shown in Tables 4.3 and 4.4 for the MMPMRs.

Table 4.3: MMPMRs @ FAR = 0.1% for Baseline Morphs (%)

Method	Score
MIPGAN [36]	78.00
Warped StyleGAN2	76.04

Table 4.4: MMPMRs @ FAR = 0.1% for Morphs using PCA (%)

Method	60/40	50/50	40/60	30/70	20/80	10/90
Element-wise Maxing	76.23	75.76	73.36	70.95	67.46	55.62
Norm Selection	76.77	75.84	75.17	73.63	70.48	67.07

From the MMPMRs, we see improved performances from both the element-wise maxing and norm selection with the 60%/40% combination. The scores decrease as the percentage that is blended using either technique increases. MMPMR takes into account both subjects used to generate the morph. As the percentages increase, so does the risk of the morph being biased toward one of the subjects. In the norm selection technique, we select which style vector to add to the average, which may result in selecting the same subject's style vector for each layer. These results show the PCA blending methods have some improvement in the morphs' performance compared to the complete averaging of the latent codes.

We compare our morphs with established morphing techniques using StyleGAN2 against FRS using a verification test and MMPMR [16, 29]. To further evaluate

Table 4.5: Results of Single-Morph Detector on FRGC Dataset

Method	APCER @ BPCER30	APCER @ BPCER10	APCER @ BPCER5
FaceMorpher [1]	2.43	12.16	18.92
Image2StyleGAN [4]	0.00	0.51	5.12
MIPGAN [36]	11.11	27.77	41.67
Warped StyleGAN2	44.67	73.81	90.86
Max 60/40	42.56	70.77	84.36
Max 50/50	38.20	71.91	82.30
Max 40/60	41.51	73.89	83.29
Max 30/70	36.10	67.53	79.22
Max 20/80	29.38	61.86	74.74
Max 10/90	29.97	53.94	68.14
Norm 60/40	51.28	81.73	91.10
Norm 50/50	46.17	73.47	85.20
Norm 40/60	44.12	76.72	91.18
Norm 30/70	43.86	81.45	95.49
Norm 20/80	45.89	83.85	93.20
Norm 10/90	54.17	80.06	94.94

the performance of our morphs, we need to evaluate them using a morph detector. We take a pre-trained FaceNet model and append a fully connected layer to serve as our morph detector model. The detector is trained on morph images generated using a landmark-based technique [1] and the StyleGAN2 warping technique used in Section 3.3 applied on the same twins dataset used to train our PCA model. These pairings allow the detector to learn what have been identified as the most challenging morphing pairs to detect [42]. The detector produces a single score to determine whether the input image is a morph or bona fide image. As the detector successfully identifies the morph images, the AUC increases while the APCERs decrease, so we aim to decrease the AUC and increase the APCERs. Performance of the morphs on the single morph detector are shown in Table 4.5 and the ROC curve in Figure 4.6.

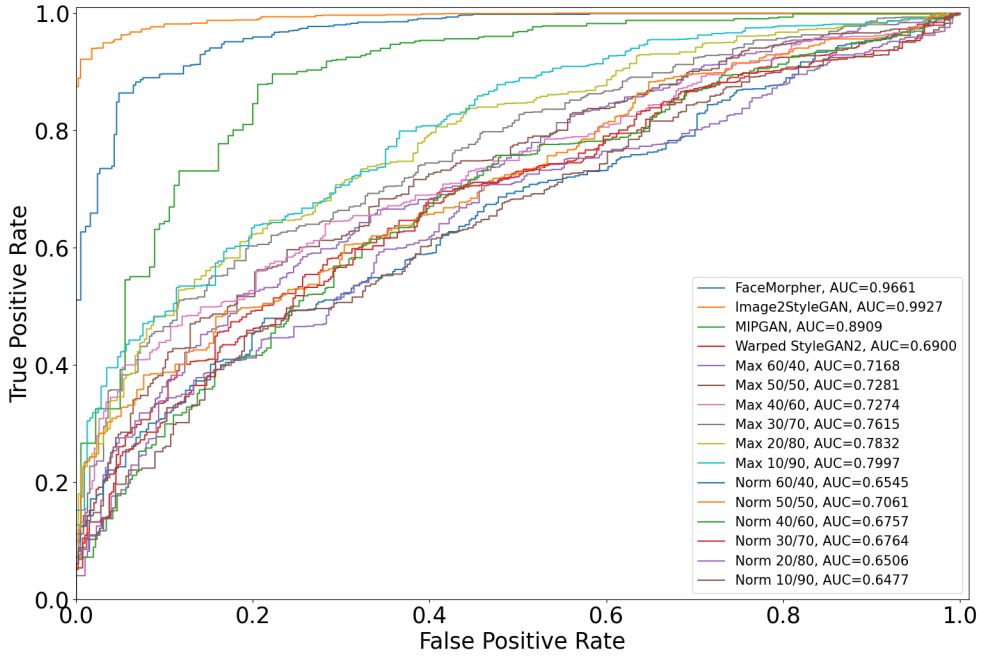

Figure 4.6: ROC curve comparing the performance of the single-morph detector on our morph techniques and previous morphing techniques.

The performance of our new Warped StyleGAN2 morphs against the single morph detector are superior to three baseline methods [1, 4, 36]. Although the model was trained on an earlier version of our landmark-warped StyleGAN2 morphs, the changes we've made have made the morphs harder to detect. The detector's performance drops as the amount of eigenvectors used in norm selection increases. Although the 10%/90% norm selection morphs perform the best on the single morph detector, we must take into consideration their reduced MMPMR and FaceNet performance, as well as visual quality.

4.3 Summary

Applying PCA to modify the blending of the styles does improve our original results in both FaceNet verification, single morph detection, and MMPMR compared to other GAN-based morphing techniques [3, 4, 36]. The variation on the performance across the three tests shows how different blending techniques can be a threat to one type of detection but a lesser threat to others. Our norm selection technique using the 60%/40% combination on our warped convex hulls performed best on average across all GAN-based morphs, improving upon the baseline results [4, 12].

Chapter 5

Conclusion

5.1 Summary of Work

GAN-based morphing poses a unique threat to FRS compared to landmark-based morphing methods due to their limited number of facial artifacts. Due to the limited number of artifacts, GAN-based morphing poses a greater threat to single-morph detectors if not trained to detect that variation of morphing. However, GAN-based morphing struggles to retain identity in the morph images, reducing their threat to FRS authentication compared to landmark-based methods. We introduced a new embedding technique, added landmark warping to GAN-based morphing, and explored alternatives to the averaging the latent codes to morph in the latent space to improve identity preservation in the GAN-based morph images. Although our GAN-based morphs do not pose the same threat to FRS as landmark-based methods, our morphs do pose a greater threat than other GAN-based morphing methods. In addition,

our morphs are unique compared to other methodologies making them ideal for deep morph detector training as they perform different than standard GAN-based morphs on single-morph detectors.

We show that landmarks do have an impact on the performance of morph images, but there is a still a loss in identity when morphing in the latent domain. Exploration into the noise input showed how embedded images can become indistinguishable from the input if allowed to learn information about the input. By limiting the noise during training, the latent representations for warped convex hulls improve their ability to learn the identity of the bona fide image. Finally, we introduced a new method of morphing in the latent domain using PCA by going deeper into the StyleGAN2 model, uncovering the similarities of style vectors across a dataset. This work is done for the pursuit of discovering limitations of GAN-based morphs as well as to develop improved and unique ways to blend latent representations in order to better understand the threat posed by GAN-based morphs to FRS and our security.

5.2 Future Recommendations

An alternative approach to blending latent representations we explored was applying one dimensional wavelets. Similar to work done by [40], we decompose two latent representations using a one-dimensional wavelet transform on each layer. The low sub-bands are then averaged while we apply element-wise maxing to the remaining sub-bands. Early test results were comparable to averaging, but this methodology lead to another potential improvement to the image inversion method. As wavelets have been proven to be effective in image morphing applications, pixel-wise loss can

be replaced with a comparison of sub-bands of the original image. We applied this method in both the image inversion and noise optimization steps with improved results. Further exploration into this could lead to an inversion method that can prioritize different frequency content of the bona fide images.

The results from training for noise after inverting an image allowed for a near perfect reconstruction of the bona fide image. The noise values, however, were not able to be blended with others to generate a morph. This could be corrected by studying further how the noise applies texture to the reconstructed image. In addition, an encoder could be added to existing image inversion methods which use an encoder to embed images [31, 32] to estimate the noise values to achieve the same texture as the input to the latent generating encoder. Although encoder-based inversion methods are not yet up to the same performance level as optimization, the performance has steadily increased within the last 2 years.

Since the beginning of this work, numerous advancements and variations have been made to the StyleGAN [2] architecture as well as new inversion methods [43, 44]. As the image generation quality improves, so will the inverted image quality. As new face generating GANs are developed, there will be a need to evaluate their morphing threat. Fine tuning the StyleGAN model with the dataset you are morphing with may also improve results as discussed by [13]. These improvements could lead to GAN-based, artifact free morphs proving a greater threat to FRS and our security.

Bibliography

[1] Quek, A (2019) Face Morpher [Source Code]. https://github.com/alyssaq/face_morpher.

[2] 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 (CVPR)*, June 2019.

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020.

[4] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4431–4440, 2019.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," June 2014.

[6] W. Wang, R. Wang, L. Wang, Z. Wang, and A. Ye, "Towards a robust deep neural network in texts: A survey," *arXiv preprint arXiv:1902.07285*, 2019.

[7] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.

[8] K. W. Bowyer, "Face recognition technology: security versus privacy," *IEEE Technology and society magazine*, vol. 23, no. 1, pp. 9–19, 2004.

[9] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[10] ICAO, *9303-Machine Readable Travel Documents Part 9: Deployment of Biometric Identification and Electronic Storage of Data in eMRTDs.*

[11] M. Ferrara, A. Franco, and D. Maltoni, "The magic passport," in *IEEE International Joint Conference on Biometrics*, pp. 1–7, 2014.

[12] S. Venkatesh, H. Zhang, R. Ramachandra, K. Raja, N. Damer, and C. Busch, "Can gan generated morphs threaten face recognition systems equally as landmark based morphs? - vulnerability and detection," in *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, 2020.

[13] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel, "Are gan-based morphs threatening face recognition?,"

[14] Luxemburg, R (2020) StyleGAN2Encoder [Source Code]. https://github.com/robertluxemburg/stylegan2encoder.

[15] A. Weaver, "Biometric authentication," *Computer*, vol. 39, no. 2, pp. 96–97, 2006.

[16] 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 (CVPR)*, June 2015.

[17] 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 (CVPR)*, June 2019.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[20] DeBruine, Lisa (2018) debruine/webmorph: Beta release 2 [Source Code].

[21] M. Ferrara, A. Franco, and D. Maltoni, "Decoupling texture blending and shape warping in face morphing," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2019.

[22] Mallick, Satya (2019) Face morph using opencv — c++ / python [Source Code].

[23] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. 60, pp. 1755–1758, 2009.

[24] J. Li, J. Yang, J. Zhang, C. Liu, C. Wang, and T. Xu, "Attribute-conditioned layout gan for automatic graphic design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 4039–4048, 2020.

[25] M. S. Mirzaei, K. Meshgi, E. Frigo, and T. Nishida, "Animgan: A spatiotemporally-conditioned generative adversarial network for character animation," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2286–2290, 2020.

[26] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," *arXiv preprint arXiv:1611.06355*, 2016.

[27] 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.

[28] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[29] 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)*, pp. 1–7, IEEE, 2017.

[30] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[31] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2287–2296, 2021.

[32] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," in *European conference on computer vision*, pp. 592–608, Springer, 2020.

[33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[34] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "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, IEEE, 2005.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] 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.

[37] 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*, pp. 1867–1874, 2014.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[40] K. O'Haire, S. Soleymani, B. Chaudhary, P. Aghdaie, J. Dawson, and N. M. Nasrabadi, "Adversarially perturbed wavelet-based morphed face generation," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pp. 01–05, IEEE, 2021.

[41] Frontex, "Best practice technical guidelines for automated border control (abc) systems," 2015.

[42] J. McCauley, S. Soleymani, B. Williams, J. Dando, N. Nasrabadi, and J. Dawson, "Identical twins as a facial similarity benchmark for human facial recognition," in *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, 2021.

[43] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[44] T. Oorloff and Y. Yacoob, "Encode-in-style: Latent-based video encoding using stylegan2," *arXiv preprint arXiv:2203.14512*, 2022.