Real vs Fake Faces: DeepFakes and Face Morphing

Jacob L. Dameron
WVU, jldameron@mix.wvu.edu

Real vs Fake Faces: DeepFakes and Face Morphing

Jacob Dameron

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

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2021

Keywords: DeepFakes, Face Morphing, Face Recognition, Facial Action Units, Generative Adversarial Networks, Image Processing, Classification.

Abstract

Real vs Fake Faces: DeepFakes and Face Morphing

Jacob Dameron

The ability to determine the legitimacy of a person's face in images and video can be important for many applications ranging from social media to border security. From a biometrics perspective, altering one's appearance to look like a target identity is a direct method of attack against the security of facial recognition systems. Defending against such attacks requires the ability to recognize them as a separate identity from their target. Alternatively, a forensics perspective may view this as a forgery of digital media. Detecting such forgeries requires the ability to detect artifacts not commonly seen in genuine media. This work examines two cases where we can classify faces as real or fake within digital media and explores them from the perspective of the attacker and defender.

First, we will explore the role of the defender by examining how deepfakes can be distinguished from legitimate videos. The most common form of deepfakes are videos which have had the face of one person swapped with another, sometimes referred to as "face-swaps." These are generated using Generative Adversarial Networks (GANs) to produce realistic augmented media with few artifacts noticeable to human observers. This work shows how facial expression data can be extracted from deepfakes and legitimate videos to train a machine learning model to detect these forgeries.

Second, we will explore the role of the attacker by examining a problem of increasing importance to border security. Face morphing is the process by which two or more peoples' facial features may be combined in one image. We will examine the process by which this can be done using GANs, and traditional image processing methods in tandem with machine learning models. Additionally, we will evaluate their effectiveness at fooling facial recognition systems.

Acknowledgments

I would like to first thank my advisor Dr. Xin Li for all his guidance, help, and kindness throughout my graduate studies; and for giving me the opportunity to work under him as a research assistant. I would also like to thank my upperclassman Ahmed Cheikh Sidiya for all his help with my research and the patience he showed when answering my many questions. I would like to thank my lab mates Shan Jia and Mindi Ruan for providing me with technical assistance and access to lab equipment during the COVID-19 lockdown. I would also like to thank Na Zhang for continuing the work on my projects after I graduate.

Further, I want to thank Dr. Natalia Schmid and Dr. Matthew Valenti for being on my committee and providing me with their advice and feedback for this work.

Finally, I would like to thank my parents for their unconditional support, sacrifice, and guidance over the years, without which this would not have been possible. Additionally, I would like to thank my brothers for their support and encouragement during my graduate studies. Most of all, I would like to thank my wife Morgan, who has shown me more love and support than words can express. Without her I would never have completed this work.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Many tools exist to digitally alter a person's face in an image or video. This is often done for entertainment purposes, whether through social media or in the film industry. However, we've seen that misusing these tools can carry a significant social risk. Thus, leading to an increase in demand for ways we can detect altered images and video. We will examine two cases of particular concern that involve digitally altering a person's face. These cases are deepfakes, which have shown a growing potential for harassment and impersonation, and face morphing which has proven to have the ability to circumvent security procedures.

1.1 Motivation

From a digital forensics perspective, the motivation of this work is to investigate the strengths and weaknesses of different forgery techniques and detection methods. From a biometrics perspective, it is to support defense efforts against potential attacks. These points are elaborated on more in the following sections.

1.1.1 Detection of DeepFakes

The popularity of convolutional neural networks (CNNs) and deep learning has soared in recent years. This has led to models that can be used to alter images and video in a way that appears realistic to the average human observer. The potential risk this can pose to an individual is compounded by the age of social media where access to someone's data is more readily available and "fake news" can propagate more easily. This increases the importance of reliable methods to detect such digital forgeries. One such method of alteration is known as deepfakes, which are primarily used to swap one person's face for another in a video clip. Our work in this area is primarily inspired by [1]'s work in detecting deepfakes of political figures using facial movement information. We seek to explore and expand on this method of detection.

1.1.2 Face Morphing vs Facial Recognition

The National Institute of Standards and Technology (NIST) has actively been conducting their Face Recognition Vendor Testing Morph[1] (FRVT MORPH) program. In their work, NIST has recognized the importance of face morphing attack and defense to organizations which issue forms of photo identification or use facial recognition for identity verification. The motivation for this work is to support research into morphing attack detection (MAD) by taking on the role of an attacker attempting to fool facial recognition systems (FRS) using face morphs. By examining this attack process, we hope to provide insight into ways to defend against such attacks.

1.2 Problem Statement

This work explores two cases where the problem can be thought of as discrimination of real vs fake human faces. First, we look at the discrimination between real and deepfake video clips. The second case is the detection of face morph attacks on facial recognition systems. In this second case however, we are playing the role of an attacker and will not directly be classifying images as morphs or not, which would be the defender's goal.

1.2.1 Facial Action Unit-Based Detection of DeepFake Videos

In this work, we classify 10-second video clips from the dataset used in [1] as one of two categories (real or fake). All real videos are genuine clips of a political figure (although we will primarily make use of clips featuring Barrack Obama), and "fake" clips are real videos of a different individual or are deepfake videos. We show how to detect fakes by training a support vector machine (SVM) on facial expression data extracted from the video clips.

We take an approach similar to [1]. However, we explore how using less data and features to train a model affects the overall accuracy. Additionally, we investigate which facial action units' interactions with one another play the most important role for determining if a video is real or fake.

1.2.2 Construction & Benchmarking of Face Morphing Attacks

In this work we construct face morphs using three methods. First, we use our own MATLAB implementation based on [2]'s "combined morph" approach. Second, we use an opensource morphing tool, slightly modified to perform splicing morphs. Third, we use a deep learning

[1] https://pages.nist.gov/frvt/html/frvt_morph.html

approach based on Generative Adversarial Network (GAN) architecture. These three approaches construct face morphs from the same set of images in the AMSL Morph dataset [2], and are benchmarked against each other and the morphs provided in the dataset.

Using the face morphs in the AMSL Morph dataset as a baseline, we take on the role of an attacker by testing each method against the VGG-Face (face descriptor) model [3]. The results show each method's ability to fool a facial recognition system into verifying the morph as its contributing subjects.

1.3 Contributions

This section provides a summary of our contributions for both examined cases of real vs fake faces.

1.3.1 DeepFake Detection

The contributions from our work in detection of deepfake videos are the following:

1. We propose to classify deepfake videos by only using facial action unit data, as opposed to the approach of using head pose information with these features (i.e. using 58 less features than [1]).
2. We classify deepfake videos using less than a third of the clips used in [1] for training (we use ~50k at most and they use ~160k). Initial testing of our classifier results in about 0.90 area under the ROC curve (around 0.05 less than [1]). We further improve this through optimization of selected training features, resulting in about 0.94 AUC when classifying deepfakes.
3. We study which facial action units are most important for detection of deepfakes for a specific person of interest. We accomplish this by changing the composition of the training and testing datasets in the "feature pruning" approach in [1].

1.3.2 Morphing Attack Defense

The contributions from our work in morphing attack defense are the following:

1. We provide a brief overview of the artifacts seen in the AMSL Morph dataset; and those seen by testing Face Morpher, our basic morphing implementation, and StyleGAN2 on the AMSL Morph Dataset.

2. We benchmark and compare the biometric quality of the morphs contained in the AMSL Morph dataset, and the morphs generated by applying each of the above methods to AMSL morph. We also show that morphing persons of different sex and/or ethnicities lowers the biometric quality of morphs. We show that the LMAs tested are typically more likely to fool an FRS, with a probability of about 0.69 in the best case, than our StyleGAN2 morphs which have a probability of about 0.55 in the best case.

1.4 Outline

In the following chapter we explore our work in facial action unit-based detection of deepfake videos. We show how to train a model exclusively on facial expression data to classify videos as real or fake. Additionally, we examine which facial action units are most important for detection of deepfakes.

In chapter 3, we explore our construction and benchmarking of face morphs. We show the results of three different tools for face morphing and compare them to the morphs contained in the AMSL Morph dataset. These morphs have their biometric quality benchmarked using VGG-Face descriptor for facial verification.

In chapter 4, we will finish this work with the conclusions drawn from our research in deepfakes and face morphing. Additionally, we will provide our thoughts on future work in real vs fake faces.

Chapter 2

Facial Action Unit Based Detection of DeepFake Video

DeepFakes are a relatively new and popular category of fake media. They are generated using Generative Adversarial Networks (GANs) to produce realistic augmented media with few artifacts visible to human observers. In the following sections we will examine the history of deepfakes and their potential societal impact, related work on the generation and detection of deepfakes, and how deepfakes can be distinguished from legitimate video using information gathered from the facial expressions.

2.1 Introduction to DeepFakes

DeepFakes are fake media generated using deep learning. The term originated when a reddit user by the name of "deepfakes" created a community on the site to share videos created from swapping celebrity faces into pornographic clips [1], [4]–[6]. This technique is commonly known as "faceswap" [1], [4], [6], and is probably the earliest and most popular form of deepfakes [4]. Since the origin of the community in 2017, the use of the term "deepfakes" has expanded from face swaps to also describe most media created or altered using deep learning[2].

In addition to swapping faces in a video, there are many other types of AI generated media that may be considered deepfakes. For example, deepfakes have been credited with mimicking voices accurately enough to fool humans and being used commit financial fraud[3]. Another common form is the so called "lip-sync" deepfakes, which changes the lip movements of a person speaking to match a new audio track [1]. And while not usually considered deepfakes, the generation of realistic synthetic faces [7], [8] may be considered a close relative when considering methods which use deep learning to produce "fake" face images. This relation is given more validity after considering methods in which these networks are used to swap the facial features of actors [4], [9]. However, for the purposes of this work, we define deepfakes as the following: Images and videos which have been altered using deep learning techniques to replace at least one actors' face or identity with another's [4], [6], [10].

[2] https://mitsloan.mit.edu/ideas-made-to-matter/deepfakes-explained
[3] https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402

The popularity of deepfakes has brought about much concern regarding their social impact and potential for misuse. This includes their potential to be used for blackmail and harassment[4], non-consensual pornography, and spoofing one's identity to provoke fraudulent financial transactions as mentioned earlier. Concern about their misuse and growing popularity has led to an increase in effort to detect deepfakes. One of the most influential examples of this push for detection methods was the DeepFake Detection Challenge[5], which offered a $1 million prize pool and shows there exists a real demand for reliable methods to detect deepfakes.

2.1.1 How DeepFakes are Made

Traditionally, the most common deep learning architectures used for generating deepfakes were variational auto encoders (VAEs) [6], [11]. Although more recently, generative adversarial networks (GANs) have become a popular model to use for generating deepfakes [1], [4]–[6], [10], [12]. While there are many nuances in the details of the many implementations of deepfake generation [4], [13], the high-level approach stays the same.

DeepFake algorithms can be broken up into two phases during use: The training phase, and the generation phase. For both phases, alignment and cropping of the input face images is the first step. These images (or video frames) will come from one of two datasets. Dataset A will contain images of the person whose face will be replaced (target identity) in some media, usually a video (target media). Dataset B will contain images of the person who will replace our target identity (source identity), in the target media.

During the training phase one autoencoder for each identity (A and B) will be trained using the datasets mentioned earlier. Autoencoders perform dimensionality reduction, which can be thought of like a type of data compression, to represent an image as a single vector. Each autoencoder consists of an encoder and decoder. The encoder will take the input image and output a single vector called the code. The decoder will take the code and reconstruct the original input image. The autoencoder is trained to minimize loss between the input and output images. A key difference in our case (when generating deepfakes) is that our two autoencoders will share weights in the

[4] https://www.bbc.com/news/technology-56404038
[5] https://kaggle.com/c/deepfake-detection-challenge

encoders. As a result, we can think of each autoencoder as having the same encoder but different decoders.

During the generation phase, the shared encoder takes in data from our target, and outputs code to the decoder for our source. Since each decoder is unique to one identity, the decoder will output a face like our source identity but with some features of the target. Additionally, the quality of the face swap is further optimized for an adversarial loss component introduced using a GAN model framework [6], [13], [14].

2.2 Related Work

While deepfakes are relatively new, there have been several published methods of detection. A large portion of these methods of detection fall into one of three categories: Raw image based, GAN specific artifact based, and physiological feature based. While these three categories are not exhaustive, they serve the purpose of illustrating a general trend in approaches to deepfake detection.

A direct approach to detecting deepfakes is through training a neural network on deepfake images and videos. An example of this technique is [6], which uses a CNN to extract image level features from video frames which are then fed to a recurrent neural network (RNN). The RNN learns to classify videos as deepfake or not based on these image level features. [6] Also mentions how autoencoders do not consider temporal information which leads to inconsistencies between video frames, a weakness of deepfakes that is exploited by this and other approaches. Another pseudo direct approach can be seen in [14], where they exploit "mesoscopic properties of images." This method could be thought of as a "fight fire with fire" approach, as it is highly dependent on deep learning methods which are used for deepfake generation.

The next category of interest is those that seek to exploit GAN specific artifacts that are introduced during deepfake generation. Theoretically, some of these approaches should also work to detect other GAN generated or altered media like those seen in [7]–[9], [15]. The method used by [10] takes advantage of the fact that deepfake methods must use affine warping to transform images after generation, due to a limit in the possible resolutions of generated images. [12] introduced a method that exploits "checkerboard" artifacts in color channels of GAN generated imagery, that are introduced during the deconvolution steps of these networks.

The last category we want to consider is those based on detection of physiological artifacts and features [1], [5], [16]. One method explored in [5] proposed classifying deepfakes based on audio-visual cues, primarily taking advantage of mismatched lip movements and voices. However, it was shown in the same work that this method performs less effective than those based on simple “baseline methods” using image quality features. When first introduced, deepfakes experienced a phenomenon where the subject in resulting video would blink infrequently or not at all. This observation was exploited in [16] and showed promising classification performance for early generation deepfakes. Modern deepfakes however, have since been improved to incorporate blinking in a more realistic way. [1] hypothesized “that as an individual speaks, they have distinct (but probably not unique) facial expressions and movements.” Therefore, mimicking a person’s appearance without properly mimicking these behaviors would be detectable as an impersonation. A weakness in this method is that it requires data from specific individuals to accurately determine their facial expression “fingerprint” to compare it with a deepfake impersonation. Therefore, this method is best suited for individuals in the public eye, such as political figures and celebrities. This method would be individual specific and unable to generally detect deepfakes. Instead, it determines if a video falls within a general pattern that is expected by a person of interest (POI). As pointed out by [1], methods which rely on these coarser (physiological) features are not as vulnerable to additive noise and recompression as those based on pixel level artifacts.

2.3 Methods and Procedure

The following section will describe our approach to detecting of deepfakes using facial action units (FAUs). It’s worth noting that this work is inspired by, and seeks to expand on, the work done in [1].

2.3.1 Dataset

The dataset used in the following experiments was taken from [1] and provided by their authors. This dataset does not include every video clip mentioned in their work, hence the discrepancies between their dataset size and ours. Here after, this dataset will be referred to as the Protecting World Leaders Against DeepFakes (PWLADF) dataset.

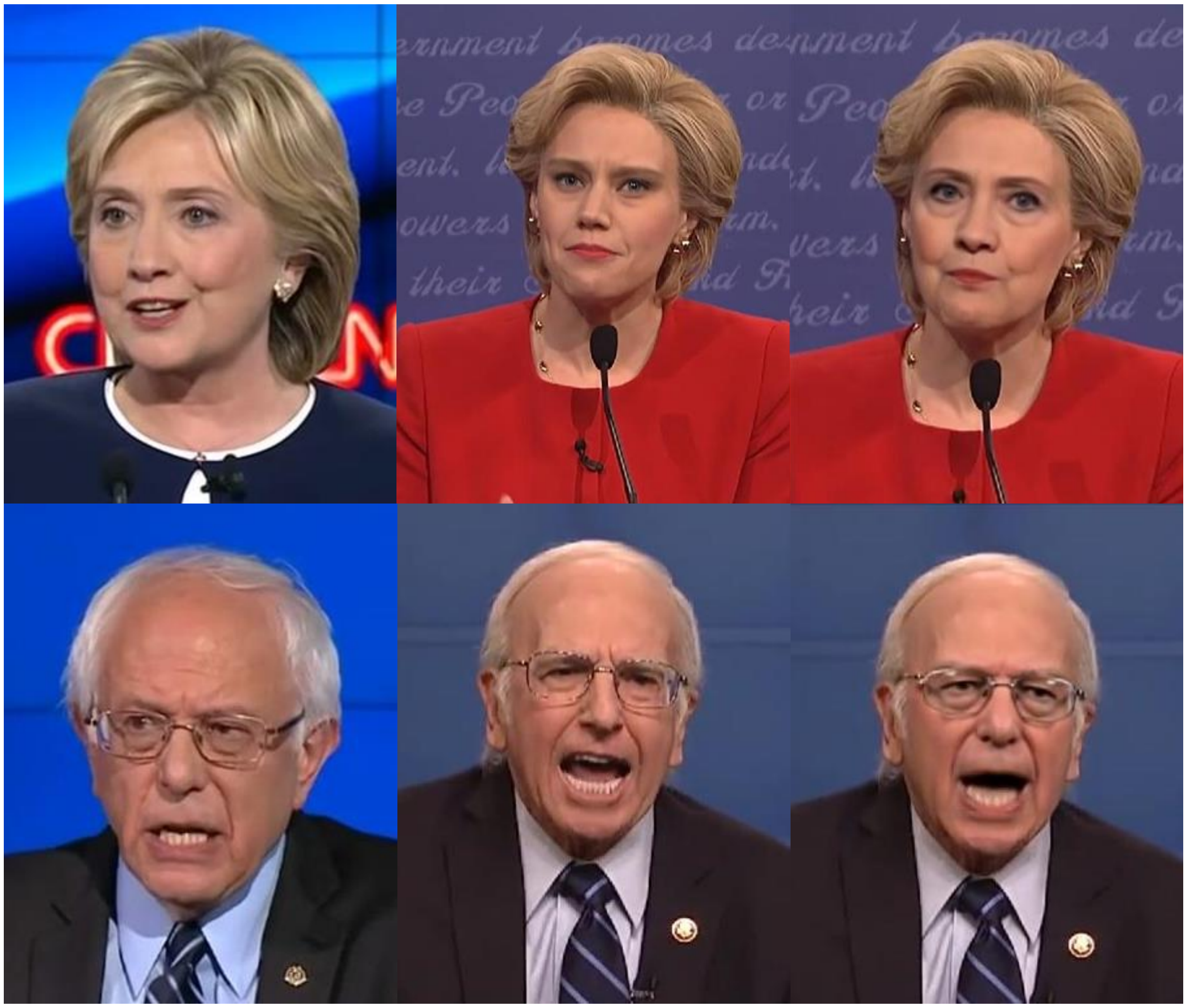

Figure 2.1: Example images taken from real video (Left) impostor video (Middle) and deepfake video (Right).

The dataset includes clips of five different political persons of interest (POIs), Hillary Clinton, Barack Obama, Bernie Sanders, Donald Trump, and Elizabeth Warren. For each POI the three categories of clips we will look at are real, impersonator, and faceswap (see Figure 2.1). Real segments come from legitimate videos of the POI and are ten seconds in length or longer. Additionally, they only contain the face of the POI, and no other individual appears in the segments. Impersonator segments come from videos containing people impersonating the POI, usually for comedic purposes. These share the same time and face exclusivity criteria as the real segments and contain no tampering by any deepfake software. The faceswap deepfakes are impersonator segments that have been altered using a faceswap deepfake tool to swap the original impersonator's face with that of the POI. The faceswap deepfakes are generated using faceswap-GAN[6]. The PWLADF dataset also includes deepfakes other than faceswap. We choose to simplify our experiments and focus our efforts on the faceswap deepfakes due to their relatively high popularity. Table 2.1 shows a summary of the section of the dataset used.

POI	Video Segments	10 sec clips
Real		
Clinton	44	15950
Obama	228	51021
Sanders	69	14932
Trump	89	20710
Warren	31	4921
Impersonator		
Clinton	6	744
Obama	20	2216
Sanders	11	1676
Trump	5	1397
Warren	7	302
Faceswap DeepFake		
Clinton	23	1949
Obama	22	1742
Sanders	5	1092
Warren	7	302

Table 2.1: Summary of PWLADF dataset

Each video segment runs at 30 frames per second, and is stored in mp4 format at a quality level of 20 [1]. Each video segment was broken up into overlapping 10-second clips. This is done by sliding a fixed 10-second long window along each segment at 5-frame intervals. Lastly, we ensured that the testing set had no overlapping frames with the training set by setting aside video segments for testing before segmenting them into 10-second clips.

[6] https://github.com/shaoanlu/faceswap-GAN

2.3.2 Facial Action Units (FAUs)

Facial action units (FAUs) [17] are basic building blocks from which we can describe emotions and complex expressions as a combination of individual facial movements. This method of encoding complex facial expressions is crucial to any kind of automatic facial expression analysis [18].

OpenFace 2.0[7] [19] is an open source toolkit "intended for facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation." Primarily, our interest in this toolkit is for extracting FAUs [18] from videos. OpenFace extracts 17 action units from video clips, the full list of action units recognized can be seen in Table 2.2 below. When processing a video, each action unit is assigned an intensity score from zero (action unit not present) to five (action unit very present) for each frame of the video.

Inner Brow Raiser (AU01)	Outer Brow Raiser (AU02)	Brow Lowerer (AU04)
Upper Lid Raiser (AU05)	Cheek Raiser (AU06)	Lid Tightener (AU07)
Nose Wrinkler (AU09)	Upper Lip Raiser (AU10)	Lip Corner Puller (AU12)
Dimplier (AU14)	Lip Corner Pressor (AU15)	Chin Raiser (AU17)
Lip Stretcher (AU20)	Lip Tightener (AU23)	Lip Part (AU25)
Jaw Drop (AU26)	Blink (AU45)	

Table 2.2: List of all facial action units recorded by OpenFace 2.0 and their corresponding AU#.

Because the FAU data is extracted for each frame, this will result in a set of time series data for each 10-second clip. Again inspired by [1], we take the Pearson correlation of each pairwise combination of facial action units over the clip. This results in a 136-element feature vector for each clip. These feature vectors will later be used to train a model to determine if video clips are real or fake. Note that for the rest of the chapter, we will refer to these correlations between FAUs as features.

To help visualize these features, we reduced the dimensionality of the feature vectors from 136d to 2d. To accomplish this, we performed t-Distributed Stochastic Neighbor Embedding using

[7] https://github.com/TadasBaltrusaitis/OpenFace

MATLAB's Statistics and Machine Learning Toolbox[8]. The results of this can be seen in Figure 2.2.

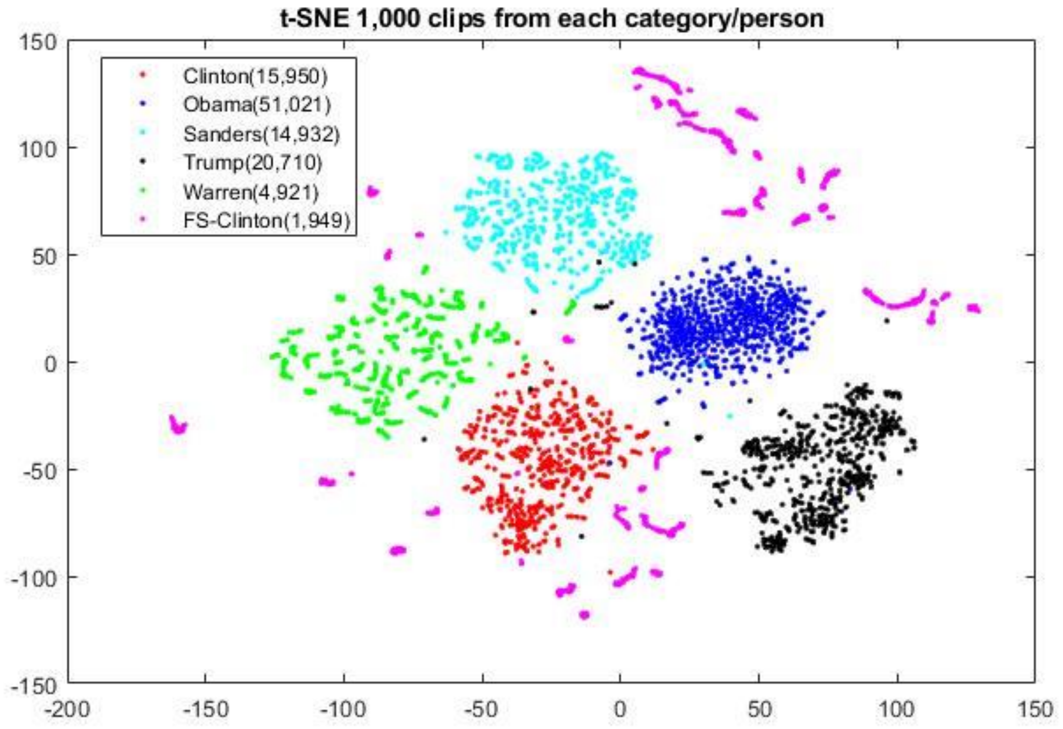

Figure 2.2: Results of using 10,000 iteration t-Distributed Stochastic Neighbor Embedding (t-SNE) on a random sample of 1,000 feature vectors from each of the following categories: Real Clinton (red), real Obama (blue), real Sanders (cyan), real Trump (black), real Warren (green), and faceswap Clinton (magenta).

As you can see, the feature vectors corresponding to the Clinton faceswap clips do not group up in the same way and are (mostly) separable from the real Clinton clips. As expected, this implies that the deepfake videos are not mimicking the pattern of facial expressions shown by a POI. Additionally, the legitimate clips for each POI tend to cluster together, implying that each POI has some facial expression pattern which distinguishes them from others. We refer to the idea of the typical facial expression patterns for an individual as a "facial expression fingerprint" (FEF). We agree with [1]'s hypothesis that FEFs are distinct, meaning they can be used for distinguishing one person from another with different patterns, and are likely non-unique to each person, unlike

[8] https://www.mathworks.com/products/statistics.html

a normal fingerprint. However, for this work we feel that this is a useful way to think about the facial expression patterns of an individual.

2.3.3 Training a Support Vector Machine (SVM)

In this section we will discuss our approach to training a classifier to detect fake videos of Obama using the PWLADF dataset. We trained a one-class support vector machine (SVM) to classify video clips as real or fake. As mentioned by [1], one-class SVMs have the advantage of only needing one class of data to train on. Additionally, they are useful for this type of problem where one class of data is less localized (fake) than the other (real). As such, one-class SVMs can be thought of as a type of outlier-detector. The outlier detection analogy is especially useful when we consider that our goal is to detect videos that lay outside what we would expect from our POIs "facial expression fingerprint." In our case, real video segments greatly outnumber deepfake video segments in the dataset. Additionally, real videos would be easier to acquire for a real-world application, especially for individuals in the public eye. For these reasons we picked a one-class SVM, which we trained using MATLAB's Statistics and Machine Learning toolbox. For training, we used a gaussian kernel with an outlier fraction of 0 and set Nu equal to 0.25. Note that a gaussian kernel is the default for a one-class SVM, and Nu controls the tradeoff between having training data fall into the positive class and minimizing the weights of the score function. These parameters were selected by trial and error and are likely not optimized.

To avoid biasing our results, we choose our testing set to be 50% real and 50% fake clips. Because the number of deepfake clips available is 1,742, and this is the smallest "fake" category, we always set aside 1,742 fake clips to use for testing. We set aside a random set of real video segments such that they can be split into at least 1,742 clips and used the rest of the data for training. Note that we throw out some data when segments are not perfectly split into 1,742 clips. Another strategy we adopted was to split our real data into 80% for training and 20% for testing, this was so we could have a "fair" comparison with the results from [1], as this is the strategy they adopt. The resulting dataset compositions can be seen in Table 2.3.

Training Approach	Total Training (10 sec clips)	Real Testing (10 sec clips)	Training:Testing (%)	Real:Fake (%)
Unbiased	49,263	1,742	93.4:6.6	50:50
80:20	40,652	10,204	77.3:22.7	85.4:14.6

Table 2.3: Dataset composition based on training approach for training a classifier for Obama. Note that the training clips are "real" clips of Obama only, as we are training a one-class SVM. The number of Obama deepfake clips available for testing was 1,742.

2.3.4 Optimal DeepFake Detection Features

Another goal of this work is to find which features are most useful for determining if a clip is real or a deepfake. Following a method outlined by [1], we train a one-class SVM on real Obama clips using only one feature (one correlation between AUs), and a smaller partition of our training data (see Table 2.4) to speed up the process. The model is then tested against faceswap Obama clips. This process is repeated for all 136 features. The feature that performs the best is saved, and then an SVM is trained in the same way using the, saved, best performing feature and one additional feature (two features total). This is repeated for the remaining 135 features. This process continues until all features are used to train the one-class SVM.

The performance of each model is measured by the receiver operating characteristic (ROC) curve which plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds. The models and features are selected to maximize the area under the curve (AUC) in the ROC plot. The AUC has a theoretical maximum of 1 and minimum of 0.5, with 1 being a perfect classifier and 0.5 being no better than random guessing. This process is visualized in Figure 2.3.

It's worth noting that in a real-world application, one wouldn't have access to the labels of the testing data. Since the AUC is optimized for the testing data, we don't necessarily consider this a general approach to detection of deepfakes, (i.e. this model likely wouldn't generalize to other data well) but rather an investigation of the data we have access to.

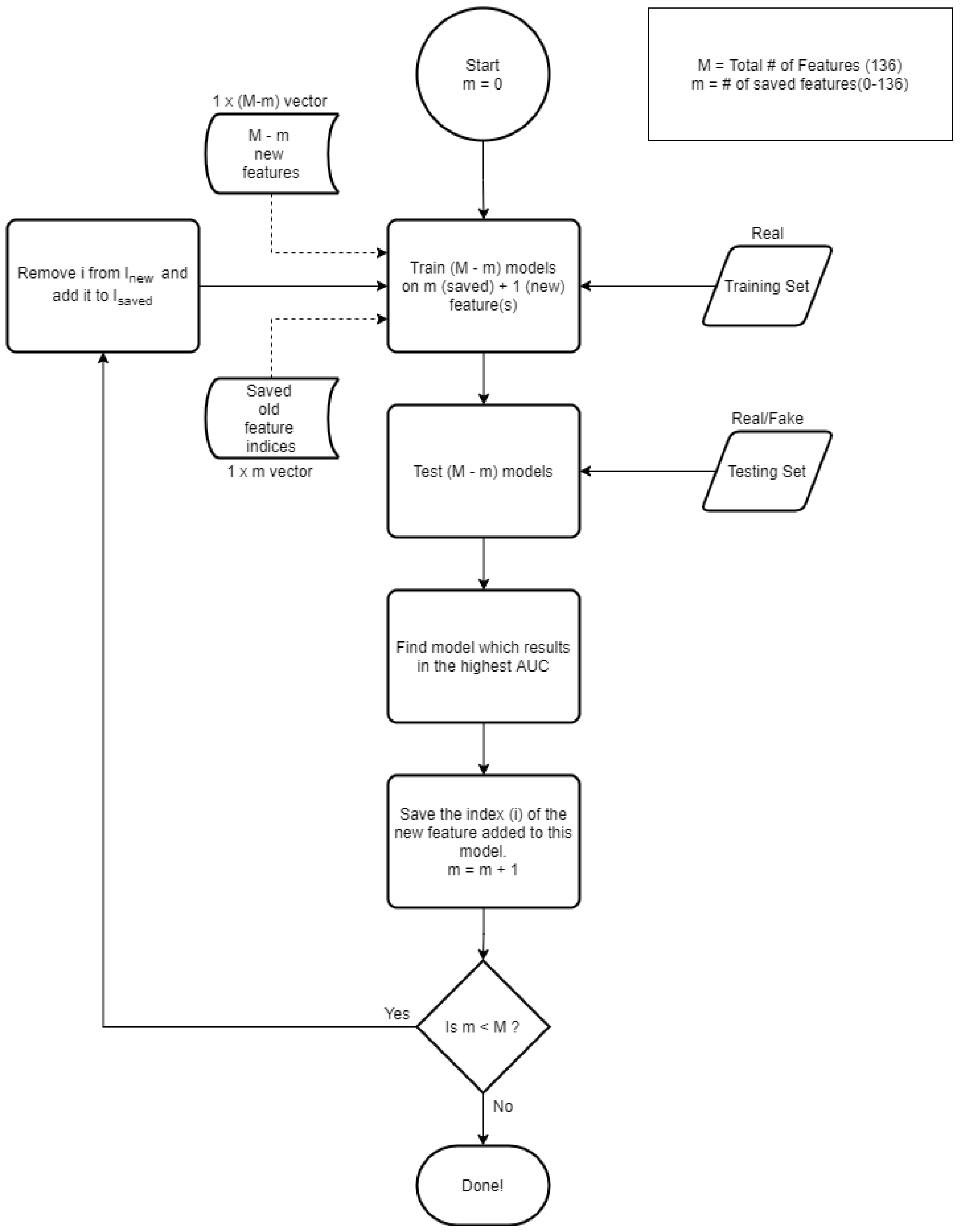
Start
m = 0
M = Total # of Features (136)
m = # of saved features(0-136)
1 x (M-m) vector
M - m new features
Remove i from I_new and add it to I_saved
Train (M - m) models on m (saved) + 1 (new) feature(s)
Real
Training Set
Saved old feature indices
1 x m vector
Test (M - m) models
Real/Fake
Testing Set
Find model which results in the highest AUC
Save the index (i) of the new feature added to this model.
m = m + 1
Yes
Is m < M ?
No
Done!

Figure 2.3: Flowchart showing the process by which we determine the “best” features for detecting Obama deepfakes.

To properly consider an optimization method, we perform another experiment following a similar method, but maximize the AUC for a “validation set” of data. The validation set is composed of 232 fake clips which do not overlap with the testing set. When computing the training (or validation) AUC, we test the model against the training set (real clips) and the validation set of fake clips. The composition of our dataset for this experiment is shown in Table 2.4..

Experiment	Total Training (10-sec clips)	Fake Valid (10 sec clips)	Total Testing (10 sec clips)	Training:Testing (Total Clip %)	Real:Fake (Testing %)
Analyze	14,184	~	3,546	80:20	50.9:49.1
Optimize	12,176	232	3,044	80:20	50.4:49.6

Table 2.4: Dataset for finding the “best” features for detecting Obama deepfakes.

2.4 Experimental Results

We tested our SVM against three different datasets: The Obama impersonator dataset, Obama faceswap dataset, and random clips from our other real datasets (random people). The performance of our model was measured by the ROC curve, which was plotted using the results from classifying the FAU data from our testing set of 3,472 (1,736 real, 1,736 fake) 10-second clips. This was repeated three times, once for each dataset standing in as our “fake” class. For each ROC curve, the area under the curve (AUC) was calculated. The results of these experiments are summarized in Table 2.5 below.

Fake Class	Impostor	Faceswap	Random People
AUC (Unbiased)	0.8955	0.9042	0.9296
AUC (80:20)	0.9024 (0.94)	0.9088 (0.95)	0.9108 (0.95)

Table 2.5: Summary of initial classification results against three “fake” classes. For comparison, we include the results in [1] shown in parenthesis.

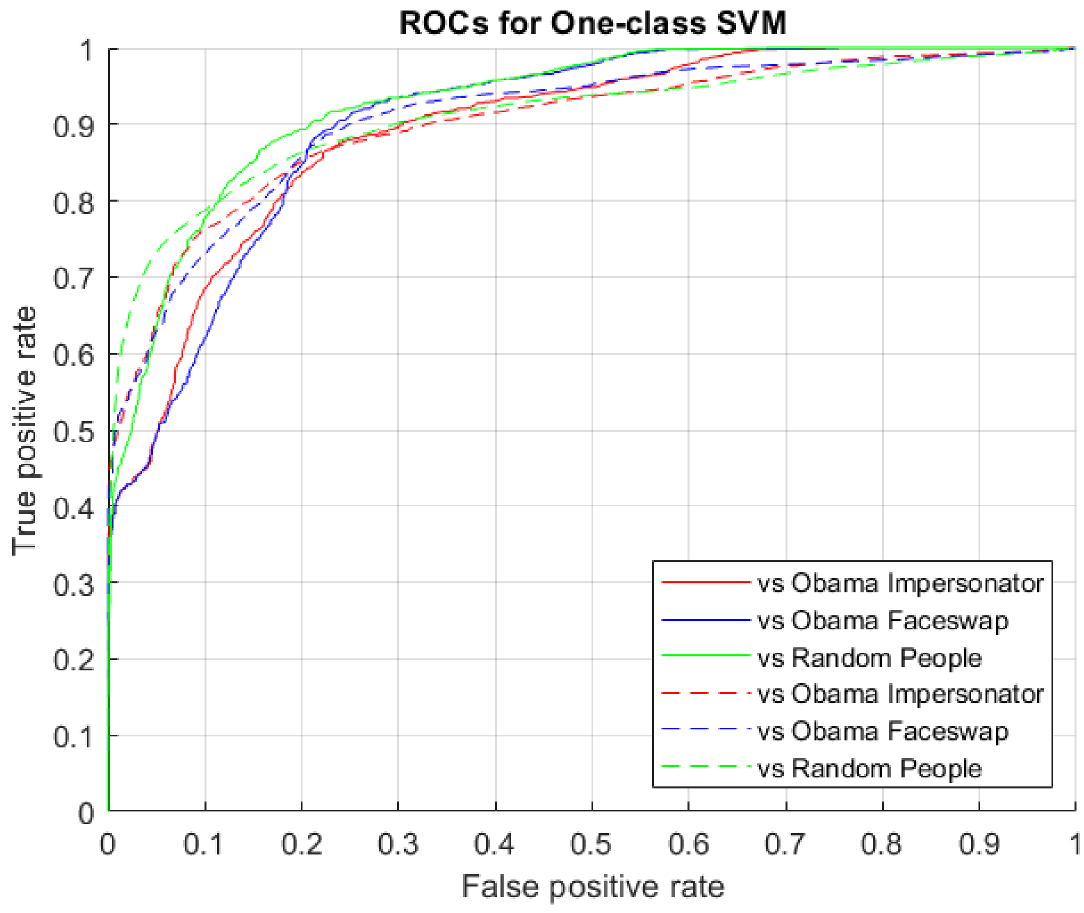

Figure 2.4: Receiver Operating Characteristic (ROC) curves for the classifiers tested against each "fake" dataset. Solid lines correspond to an SVM trained using the "Full-Set", and dashed lines correspond to the "80:20" set.

Overall, our model performed reasonably well with an AUC around 0.90 for all cases. The model was best at detecting a difference between our POI and other random POIs. Lending validity to the concept of a "facial expression fingerprint." This shows that it's possible to greatly reduce the time required to collect data (~7 times fewer clips) and use less features (no head pose information used) while sacrificing little classification accuracy compared to [1] (~0.4-0.5 AUC). We would like to remind the reader that these results may be further improved through optimizing the parameters mentioned at the start of section 2.3.3.

2.4.1 Model Optimization

Below shows the training and testing AUC as a function of features added. The training AUC is calculated by testing the model against the training set and the set of fake clips we set aside for validation. As you can see in Figure 2.5, the training AUC increases sharply at first, levels off, and then begins to decline. Therefore, by using the first couple of features we can theoretically improve the classification accuracy, and significantly simplify the model. The testing performance peaks at 17 features with an AUC of 0.9667. The training performance peaks at 21 features with an AUC of 0.9995. At 21 features the testing AUC would be 0.9446, this would be the real-world AUC achieved by our model as we would have no way of knowing that 17 features work better for our testing set. Figure 2.6 is a zoomed in version of Figure 2.5, to show the first 15 features on the x-axis.

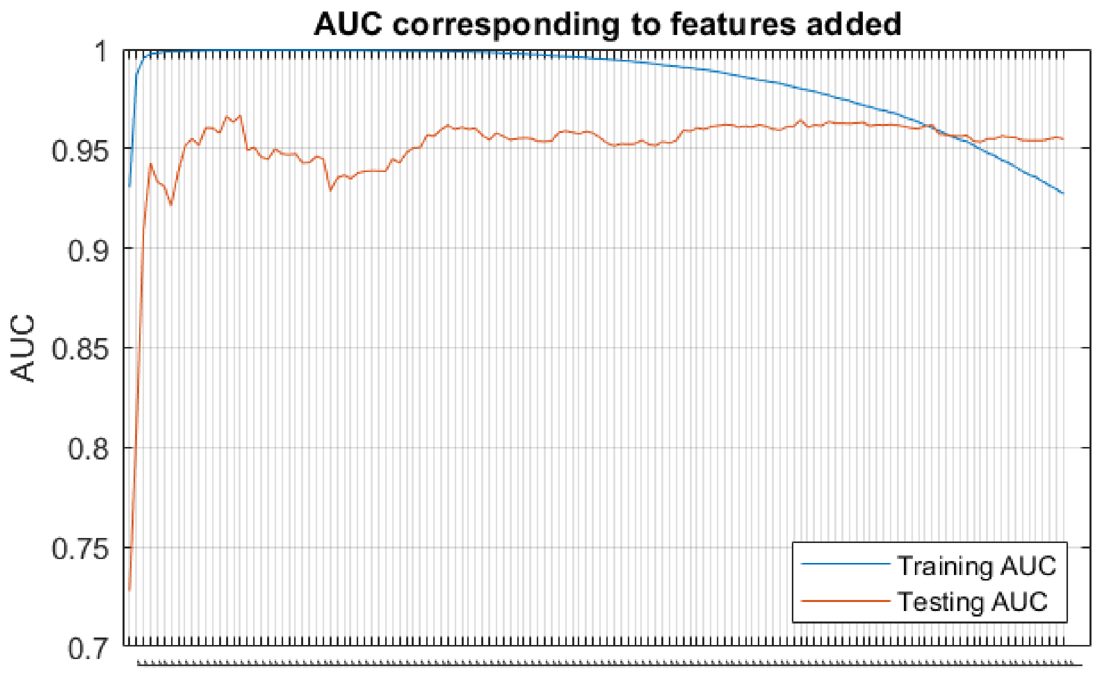

Figure 2.5: AUC as a function of features used for training. The first point corresponds to the best performing (in terms of AUC) classifier trained on a single feature. The last point corresponds to a classifier trained on all features.

Note that the performance could be further improved, and model further simplified, by using the model which corresponds to the first AUC gain seen below a certain threshold (i.e. the first model which improves the AUC by less than 0.01). We conclude that this method of model optimization provides a moderate improvement to our classification accuracy when detecting deepfakes.

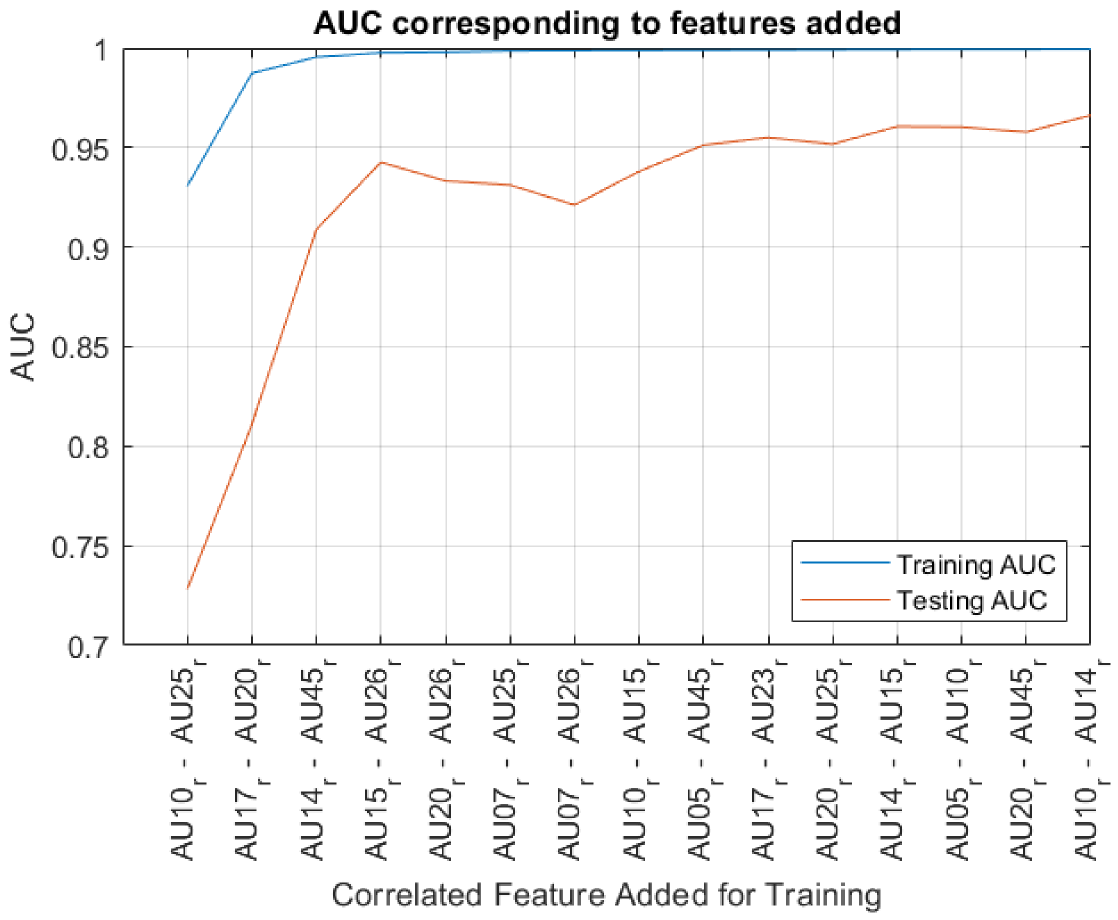

Figure 2.6: AUC as a function of features added. Training performance levels off well before feature/iteration 15.

2.4.2 Feature Analysis

The following contains the results of our feature analysis experiments described in section 2.3.4. The curve below shows the resulting AUC as a function of the number of features used for training.

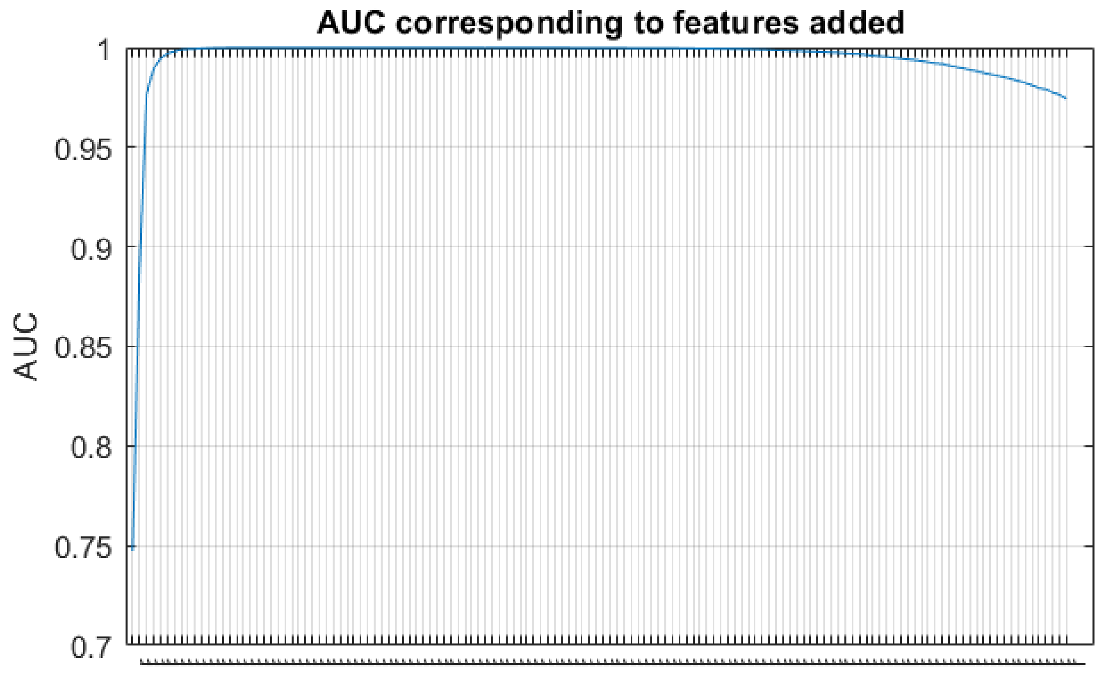

Figure 2.7: Testing AUC as a function of features used for training. The first point corresponds to the best performing (in terms of AUC) classifier trained on a single feature. The last point corresponds to a classifier trained on all features.

We again see, the AUC increases sharply at first, levels off, and then declines. Figure 2.8 below shows the same graph as above but zoomed in to show the first 15 features on the x-axis. The performance peaks at 15 features with an AUC of 1, the performance of a perfect classifier. This is a noticeable improvement over using all 136 features (~0.90), however these features are optimized for classifying the testing set which is likely an unfair comparison. To better understand how the features rank in determining if a clip is real or not, we show the top five features for the first three iterations of this experiment in Table 2.6 below. Note that the increase in AUC for each feature levels off heavily after 3 features. It's also worth noting that these rankings are when testing against Obama faceswap clips and may change for different POIs.

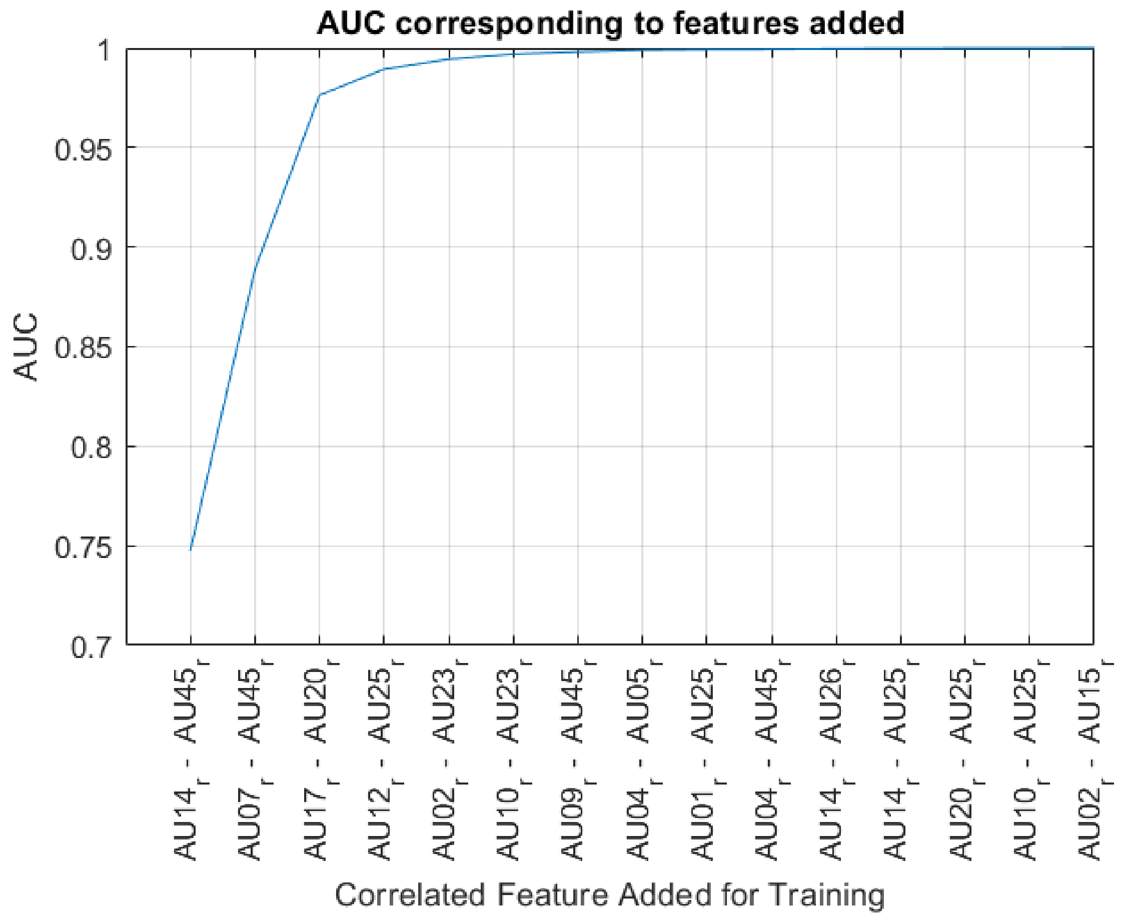

Figure 2.8: Testing AUC as a function of features added. Reaches theoretical maximum at feature/iteration 15.

A high-ranking feature indicates that it is a useful measure to check for our real POI, for one of two reasons. Either this feature represents a manor of expression that is difficult to mimic by an impostor, or it is altered through the deepfake process. It is easy to understand why a feature would be useful for detecting deepfakes in the latter case. In the former case, it would also be useful as any deepfake process requires an impersonator on some level to place the face of our POI over. Therefore, a higher-ranking feature corresponds to a more useful feature for detecting deepfakes of our POI.

# Feats/ Iterations	Feature Identifiers					AUC with that Feature added				
	Best	2nd	3rd	4th	5th	Best	2nd	3rd	4th	5th
1	14-45	10-15	07-45	20-23	10-25	0.7475	0.7318	0.7099	0.7081	0.7024
2	07-45	20-25	09-45	17-20	09-14	0.8886	0.8827	0.8812	0.8762	0.8741
3	17-20	10-15	20-25	02-23	10-25	0.9763	0.9728	0.9719	0.9643	0.9622

Table 2.6: Top five best features for the first three iterations of our feature selection process. Note: 14-45 means the feature corresponding to the correlation of AU14 and AU45.

Looking at the five best performing features on the first two iterations, one notices that the AUs dealing with the eyebrows are not present (AU01,02,04). Likely indicating that eyebrow movement is not a good indicator of deepfake videos in general, or not important to the FEF of the POI. The other AUs which do not appear in the first three iterations are Upper Lid Raiser (AU05), Cheek Raiser (AU06), Lip Corner Puller (AU12), and Jaw Drop (AU26). We also notice that correlations with AU45 (Blink) are the best performing features for iteration one and two. We suspect that this is one case where the AU can represent an artifact of deepfakes in general, due to the history deepfakes have with blinking artifacts [16].

One might expect that the second best performing added feature would be the best on the following iteration, and the third would become the second and so on. However, we don't see this. For example, an SVM trained on the Pearson correlation of AU10 and 15 (10-15) is the second-best performing model on iteration one. But, adding 10-15 to the training doesn't appear in the top five for iteration two. This implies that while 10-15 is one of the most useful features for this POI, some of the information carried by 10-15 is also carried by 14-45.

While it's tempting to generalize from these results to claim which FAUs are most useful for detecting deepfakes, we must consider a few things first. The rankings of these features would likely change for a different POI. As we've seen in the t-SNE plot, each POI can be separated based on the FAU information. And as explained earlier; either this feature represents a manor of expression that is difficult to mimic by an impostor, or it is altered through the deepfake creation process. And while both are useful for detecting a deepfake, the former case would be POI specific.

2.5 Summary

It is possible to misuse deepfake technology for nefarious purposes such as black mail, fraud, and political manipulation. As the technology improves, fakes will become more convincing and harder to detect. Because of this, it is important to develop methods which can be used to detect deepfakes.

In this work we explored the role of a defender by examining how deepfakes can be detected using an SMV trained on facial action unit (FAU) data. Additionally, we examined which features were most useful to a classifier for detecting deepfakes of a specific person of interest (POI).

While this work was inspired by [1], there are some subtle differences between the works that should be acknowledged. First, we exclude using information about head pose when correlating features. This means that the only information used for classifying deepfake videos are facial action units. Second, this work used significantly less real video segments in our dataset (for Obama; [1] used ~200k real 10-second video clips, and we used ~51k). We implemented a method to specifically find which features are most useful for detecting deepfakes. We also looked at the top five performing features over the first three iterations of this process. While we anticipate that a model trained in this way would not generalize well to other data, we find this to provide insight into the “best-case scenario” for the detection of deepfakes.

Chapter 3

Construction & Benchmarking of Face Morphing Attacks

Face morphing is a process by which one can combine the facial features of two (or more) individuals into one image. It has been shown that facial recognition systems may recognize such face morphs as multiple identities which contributed to the morph. Especially in the case of Automated Border Control (ABC) systems, this can be dangerous in the hands of a malicious actor. As displayed by [20], this allows one to bypass ABCs without alerting authorities to their true identity. In this chapter, we will examine the role of an attacker trying to fool facial recognition systems by morphing the faces of two individuals.

3.1 Introduction to Face Morphing

A face morph is an image generated from images of different faces. The idea is to take two (or sometimes more) genuine (also called "bona fide") images of different individuals and combine them into an image that resembles the faces of both contributing subjects. While this can be used for entertainment purposes, it can also be misused to exploit weaknesses in facial recognition systems.

Automated facial recognition systems (FRS) vulnerability to face morphing attacks was first investigated by [20]. The process by which this can be exploited is well explained by [20], [21]. In the case of Automated Border Control systems (ABCs), it is possible for an attacker (or criminal) to pass through a checkpoint without being recognized as the person they are. They would do this by morphing a genuine image of themselves and an accomplice. The accomplice then would apply for an ePassport. In many countries the image used for an ePassport is provided by the applicant [21], so they may use the morphed image as their photograph. Most ABCs will scan an electronically Machine-Readable Travel Document (eMRTD) or passport and check the photo on the document against a probe image taken at the gate or checkpoint. Since the morphed image resembles both the attacker and accomplice, the gate will recognize either person as the image on the passport. This allows the attacker to pass through an ABC gate while appearing to be the accomplice, and thus not setting off any red flags in the case that the attacker is wanted by the authorities.

3.2 Related Work

3.2.1 Landmark Based Morphing Methods

As already mentioned, [20] was the first to meaningfully investigate face morphing as a method of attack against facial recognition systems. The traditional way to go about face morphing is by warping faces such that key facial landmarks align in a particular way. These so-called landmark-based morphing attacks (LMA) follow a standard pipeline in most works. This pipeline contains three key steps according to [21], which are Correspondence, Warping and Blending.

The correspondence step involves defining landmarks on the face images to be morphed. These landmarks correspond to key points on the face that define its shape and structure (eyes, nose, mouth etc.). Landmarks can be defined manually [20], or more commonly they are detected automatically. One of the most frequently used [2], [22], [23] models for this is the dlib landmark detector [24].

The next step is warping of the correspondence points defined earlier. The current state of the art way to do this is by applying Delaunay triangulation and then warping each triangle in the resulting mesh. The goal of warping is to adjust the image such that corresponding points (landmarks) align between the images. Each image is aligned according to some factor α (between 0 and 1), which defines how much each subject contributes to the morph. For example: An α equal to 0 means image B will be warped to image A's landmarks, 1 means image A will be warped to image B's, and 0.5 means both will be warped to a set of points halfway between the points of the images. Naturally, to produce a morph that is similar to both faces an α of 0.5 makes the most sense [2], [22], [23], [25].

The final step is blending, which forms the morph as the weighted average of the two aligned (warped) images. At a weight of 0.5, this is the same as averaging the two images. The weight used for each image is traditionally the same as α defined earlier [2], [22], [23], [25]. However, [26] has shown promising results by separating α into two different variables for warping and blending (α_W, α_B).

For this work, it's worth noting the following variations in the landmark based approach to face morphing. The typical approach described above could be considered a "complete morph" [2], [22], [23], since the entire image is morphed. An alternative to this is described as "splicing morphs" where only the facial region is morphed, and then spliced onto the facial region of one of the contributing (genuine) images [22], [23]. This however can result in artifacts along the border of the splice as noted by [2]. As an extension of this [2] proposes "combined morphs," which instead align the images before warping. Alignment is commonly done such that the eyes of both images overlap when superimposed as described in [20], [27].

3.2.2 Deep Learning (GAN) Based Morphing Methods

In addition to LMA, deep learning approaches have also become a popular choice for constructing face morphs. [25] proposed a generative adversarial network (GAN) which could be used to construct face morphs (MorGAN). However, [28] showed that it was outperformed by using StyleGAN [7] due in part to MorGAN's resolution limitations. [15] showed an effective way to import images into StyleGAN and its applications to image morphing. Because StyleGAN (and StyleGAN2 [8]) has no direct image input, images must be embedded into the "latent space" (W) of the network. [15] further improved the morphing applications by proposing an "extended latent space" (W+) to embed the images into.

3.2.3 Benchmarking Methods

[2] proposed to measure the quality of face morphs along three different axes: Visual quality, forensic quality, and biometric quality. A morph with a high visual quality would have few visible artifacts and would appear similar to both contributing individuals. A morph with high forensic quality would have few traces of illegitimate image tampering. And a morph with high biometric quality would often be matched with the contributing identities by a facial recognition system. The biometric quality was measured in terms of FRR (false rejection rate), and MAR (morph acceptance rate) or rMAR (realistic MAR) at a FAR (false acceptance rate) of 0.1%. Each of these metrics are explained more in section 3.3.3.

When testing face matchers, [2] had shown that the VGG-Face descriptor [3] produces results consistent with a commercial face matching software Luxand face SDK[9] and, in fact, was slightly more robust against morphs.

[9] https://www.luxand.com/facesdk/

3.3 Methods and Procedure

By examining face morphing from an attacker perspective, we hope to provide insight which can be used to better defend against them. We confine our investigation to (mostly) automatic ways to produce face morphs, and thus ways to produce and improve large datasets and support research into defense against face morphs. As we are taking on the role of an attacker, our efforts will primarily be focused on fooling facial recognition systems (FRSs).

3.3.1 Dataset

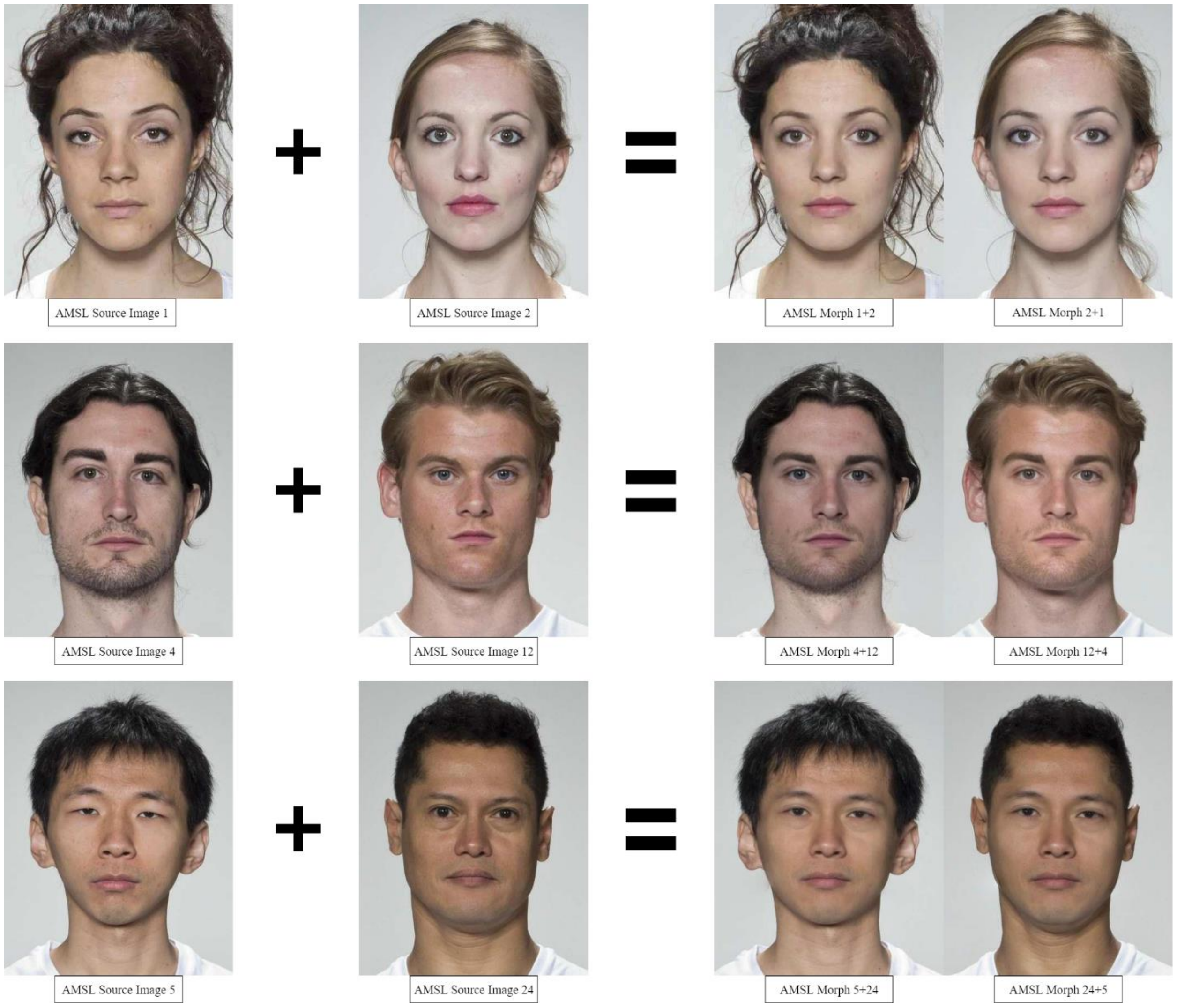

Figure 3.1: Examples of morphs from the AMSL Morph Dataset.

The AMSL dataset[10] [2] consists of a set of 102 genuine images of persons displaying neutral expressions, 102 genuine images of persons smiling, and 2,175 LMA face morphs. Examples of this dataset can be seen in Figure 3.1. Their set of genuine (non-morphed) images are from the Face Research Lab London Set[11] [29] and the Utrecht ECVP Dataset[12]. Both the neutral expression and smiling images are front facing images of the same 102 individuals. The images are modified to be 531x413 resolution, are JPEG compressed with a variable quantization factor to be below ~15kB, and comply with ICAO eMRTD standards [30]. The face morphs are created using pairs of the 102 neutral expression images and the morphing method outlined in [2]. The morphs are constrained such that only subjects that share the same sex and ethnicity are morphed together. Additionally, each morph is a “Half-way” morph (i.e. $\alpha = 0.5$), so both subjects contribute an equal amount to the morph.

3.3.2 Morph Attack Construction

As mentioned in section 3.2, there are two basic approaches to generating face morphs. First, is the traditional way of using facial landmarks, triangulation, and image warping, these are commonly called landmark based morphing attacks [25] or landmark aligned [31] (LMA) morphs. Second is the deep learning approach which uses generative adversarial network (GAN) architecture to generate morphs. We will walk through our use of each of these methods.

For landmark morphing attacks, we created our own MATLAB implementation based on the “complete morph” method outlined in [2]. Our process is explained in Figures 3.2-3.3. The first part of our process is to detect 68 facial landmarks using dlib [24] landmark detector. We use Yuval Nirkin’s find_face_landmarks library to easily interface the landmark detector with MATLAB[13]. As mentioned by [23] most “ID photographs require a closed mouth,” and this can lead to having colinear landmarks. To prevent this, we take a similar approach to theirs and remove 3 landmarks outlining the bottom contour of the upper lip.

The next step is to create versions of both images that are aligned to the other original image. To align images, we first form landmarks for the left and right eyes of each image. This is done by taking the average of the landmarks around each eye to approximate the center of the iris. The

[10] https://omen.cs.uni-magdeburg.de/disclaimer/index.php
[11] https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666
[12] PICS, see http://pics.stir.ac.uk/
[13] https://github.com/YuvalNirkin/find_face_landmarks

next steps are to then rotate the image until the head is at the same angle, resize the image such that the distances between irises are the same, and circular shift the image until the irises finally overlap between the two images. While alignment of the faces is obviously necessary to produce morphs. An additional reason to create two versions of alignment is so that we can easily splice the resulting morphs onto the original images (i.e. for any size image).

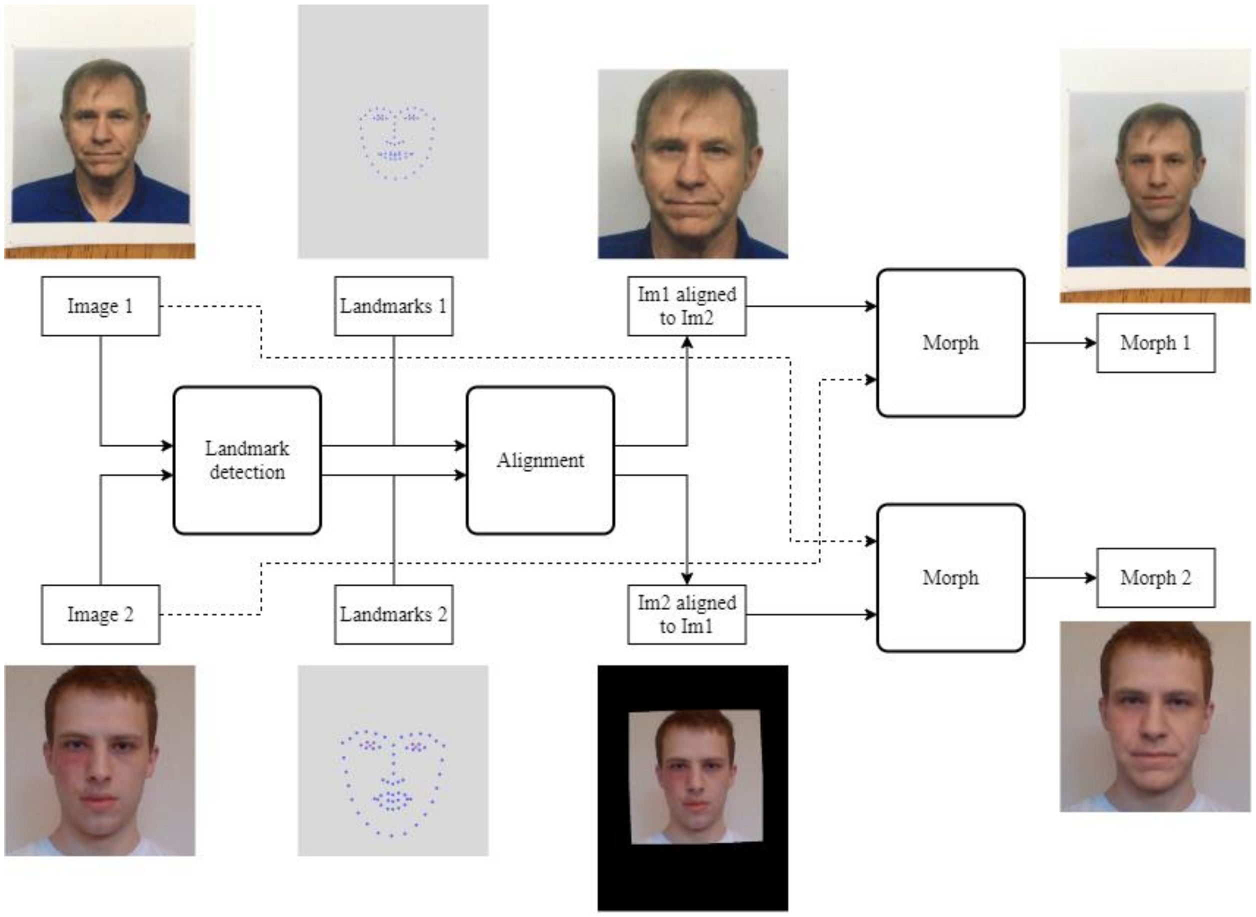

Figure 3.2: Block diagram describing alignment process. Mean eye landmarks (red) are used for alignment.

After alignment we mostly follow the standard pipeline for LMA morphs. We take the weighted average of each corresponding landmark to create a set of landmarks “in-between” both original faces images. These averaged landmarks, and the original ones, then go through Delaunay triangulation and we warp each triangle in the contributing images to its corresponding triangle in the averaged landmarks. We then cross dissolve, or average, the two warped images producing our face morph.

The next step is to splice our face morphs onto the original contributing images. This is done using the Poisson image editing process from [32]. The result is two images with the same face morph blended onto both contributing images.

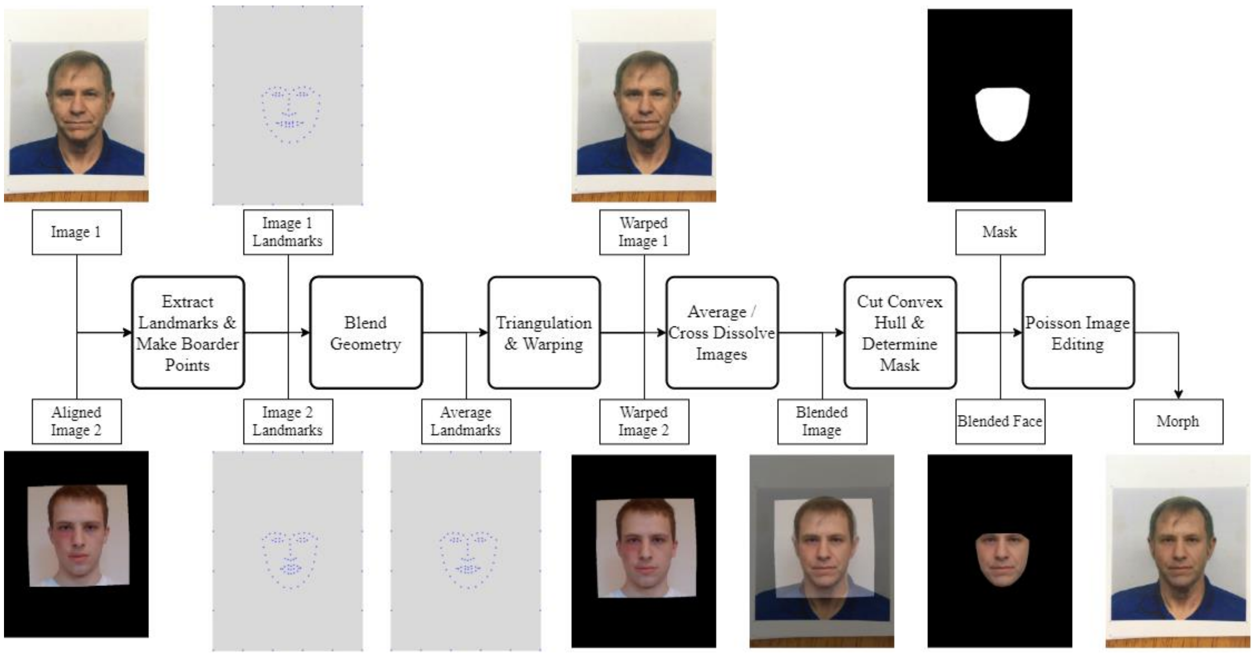

Figure 3.3: Morph pipeline describing our MATLAB implementation's morphing process.

In addition to our own LMA implementation, we also make use of a popular opensource LMA python implementation. Alyssa Quek's face_morpher[14] has also been examined by [21], [33] and is publicly available on GitHub. As with most LMA tools, it follows the standard morph pipeline and uses the dlib [24] landmark detector. [21] points out that the outer region (background) is unavailable after morphing when using this tool. We address this drawback by modifying Face Morpher to allow for splicing onto the background using Poisson image editing.

To generate GAN based morphs we make use of StyleGAN2 [7], [8] which has become one of the standard networks for GAN based face morphing. Our process for generating GAN based morphs is shown in Figure 3.4. As mentioned earlier; to use StyleGAN (1 or 2) for morphing, we must first project the source images into the "latent space" of the network. Before projection, we align[15] images to the FFHQ [7] dataset that StyleGAN2 is primarily trained on. This resizes the images from 413x531 to 512x512. We project images into the StyleGAN2 latent space using

[14] https://github.com/alyssaq/face_morpher
[15] https://gist.github.com/lzhbrian/bde87ab23b499dd02ba4f588258f57d5

Learned Perceptual Image Patch Similarity (LPIPS) [34] as a metric for perceptual similarity. This will encode the input images we want to morph into the "extended latent space" $W+$ [15]. The traditional latent space (W) is a 512x1 vector that feeds to each layer of StyleGAN2. As described by [15], $W+$ is a 512x18 matrix which feeds a 512x1 vector to 18 different layers of StyleGAN2.

The final step for producing StyleGAN2 based morphs is to take the average latent code of the projected input images and feed it to our generator. The output is a "synthetic" face that is forced to look like both of our contributing identities by using a latent code that is the average of the projected images' latent codes. It's worth noting that StyleGAN2's architecture is based around producing high resolution images. As such, all images from StyleGAN2 are 1024x1024 resolution. This contrasts with our LMA methods which will produce the same resolution images as AMSL Morph, which is 413x531.

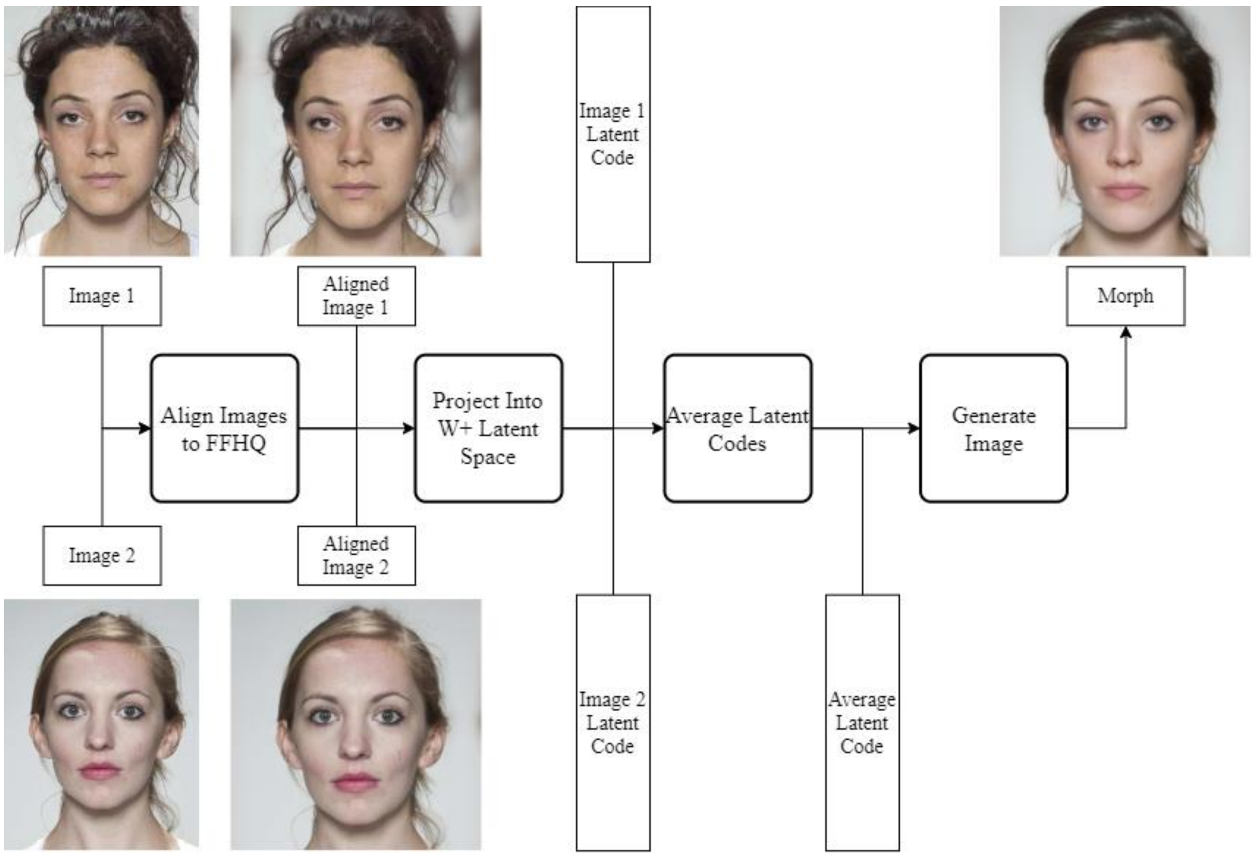

Figure 3.4: Morph pipeline describing the StyleGAN2 morphing process.

3.3.3 Benchmarking

As mentioned in section 3.2.3, face morphs can be measured by their biometric, visual, or forensic quality. We will borrow these concepts from [2] as a way to describe the quality of a morph. For this work, we are primarily interested in the biometric quality of face morphs as it dictates the ability to fool automated face recognition systems (FRS).

Any FRS can be adjusted by changing its required matching threshold (τ) which dictates the minimum similarity score needed for an FRS to match two images. [20] points out that for ABC systems it is desirable to have a false acceptance rate (FAR) no greater than 0.1%. Because this (ABC systems) is considered the most real world applicable scenario, it is common practice when benchmarking morph attacks against FRSs, to adjust the matching threshold such that the FAR is 0.1% [2], [22], [23].

We will use two metrics to describe the biometric quality of the morphs in this work. First is morph acceptance rate (MAR), which was proposed as a metric to evaluate morph quality in [35]. The basic concept is that when a morph matches with one of its contributing images (i.e. its similarity score is greater than τ) it is considered a successful morph trial. If a morph does not match with one of its source images (its similarity score is less than τ) it is not considered a successful morph. In our case, all morphs are constructed from two contributing images. Therefore, any morph will have two similarity scores which dictate whether it matches with the contributing identities or not, and the total number of morphing trials will be two times the number of morphs. The MAR is calculated as the total number of successful morph trials divided by the total attempted morph trials and is a function of τ as described in Equation 3.1 [22].

$$MAR = f(\tau) = \frac{\#accepted\ morphing\ trials}{\#total\ morphing\ trials} \tag{3.1}$$

The other metric we will use to describe the biometric quality of morphs is the realistic morph acceptance rate (rMAR), as proposed by [23]. This metric is like the MAR, except that a morph must match with its contributing images, and any other images of those identities, for it to be considered a successful morph. The rMAR is determined to be a function of τ, and is calculated as the total number of successful morphs divided by the total number of morphs as described in Equation 3.2 [22].

$$rMAR = f(\tau) = \frac{\#successful\ morphs}{\#total\ morphs} \tag{3.2}$$

Naturally, we expect the rMAR to be much lower than the MAR as it has more conditions for a success. We will evaluate both at a threshold that yields an FAR of 0.1%, and report the constant FRR for the system as also done by [2].

As mentioned in section 3.2, [2] had shown results consistent with commercial face matchers when using the VGG-Face CNN descriptor described in [3] as an FRS. To act as our FRS, we take inspiration from [2] and make use of the same VGG-face descriptor's[16] MatConvNet implementation. The network architecture is designed to take a 224x224 color image as the input. The last layer of the network is removed such that the output is a 4096-dimensional feature vector, instead of a set of classification scores. Each image is passed through face detection and segmentation using the cascade object detector from MATLAB's computer vision toolbox[17]. The bounding box of the face is extended evenly in each direction such that its size is increased by 20%. Each image is then cropped according to the extended bounding box and resized to be 224x224. The images are run through VGG-Face, and the resulting feature vectors are L2 normalized to have unit length. After processing all images, the pairwise Euclidean distance is calculated between all feature vectors. These distances are then normalized to be between zero and one by dividing by the greatest distance. The similarity scores between all images is then calculated to be one minus the normalized distances, such that similarity scores have a range from zero to one (one being most similar, zero being least similar).

3.4 Experimental Results

Using the AMSL Morph dataset as our starting point, we experiment with different methods to produce face morphs. The AMSL Morph dataset is particularly useful for our purposes because it contains two frontal facing images for each identity (neutral and smiling), which we will see is useful for benchmarking later, and because it contains a set of face morphs we can use as a baseline. It should be noted that all morphs in the following experiments are formed from neutral expression images in the AMSL Morph dataset.

[16] https://www.robots.ox.ac.uk/~vgg/software/vgg_face/
[17] https://www.mathworks.com/products/computer-vision.html

3.4.1 Morph Dataset Generation

For generating datasets, we made use of three methods to generate face morphs. Each method of morphing plays a specific role in aiding our understanding of face morphs and their capabilities. First, we crafted our own MATLAB implementation for producing landmark based morphs. This allowed us to gain firsthand insight into the process of face morphing at a low level. Second, we made slight modifications to Face Morpher, which is a popular opensource python and dlib based face morphing tool. This tool is fairly easy to setup for anyone with a little python experience, and we think of this as being the typical tool available to most malicious actors since it has a relatively low barrier to overcome for use. Third, we used StyleGAN2 [7], [8] to generate morphs from the latent codes of projected contributing images. As mentioned earlier, the two basic categories of face morphs are LMA and GAN based morphs. Since StyleGAN and StyleGAN2 are arguably the most popular GANs for face morphing, we think this serves as a good method to represent GAN based morphs for comparison against our two other LMA methods. Each of these methods is explained further in section 3.3.2.

For each method every pairwise combination of contributing images was morphed. Since AMSL has 102 neutral face images, the total number of morph combinations is given by:

$$\binom{102}{2} = 5{,}152 \tag{3.3}$$

This is the total number of StyleGAN morphs we produce. For LMA morphs, we splice the morphed face region onto the background of one of the contributing images. Since each morph has two possible backgrounds (one for each contributing image), we produce two images from each pair. Therefore, we produce 10,302 morphed images (or two times the number of pairwise combinations) for LMA methods. As mentioned earlier, the AMSL morph dataset does not contain morphs between people of differing ethnicity or sex. As a result, their total number of morphs are much lower than the amount we produce, which is not constrained by differing sex or race between subjects. The resulting set of morphs are summarized in the table below (Table 3.1). It's worth noting the resolution of all images are 413x531 except for the StyleGAN2-based morphs which are 1024x1024.

Method/Dataset	Morph Type	Resolution	Total Morphs
AMSL Morph Dataset	LMA	413x531	2,175
Our MATLAB Implementation	LMA	413x531	10,302
Face Morpher	LMA	413x531	10,302
StyleGAN2 Morph	GAN	1024x1024	5,152

Table 3.1: Summary of the morphing attacks constructed.

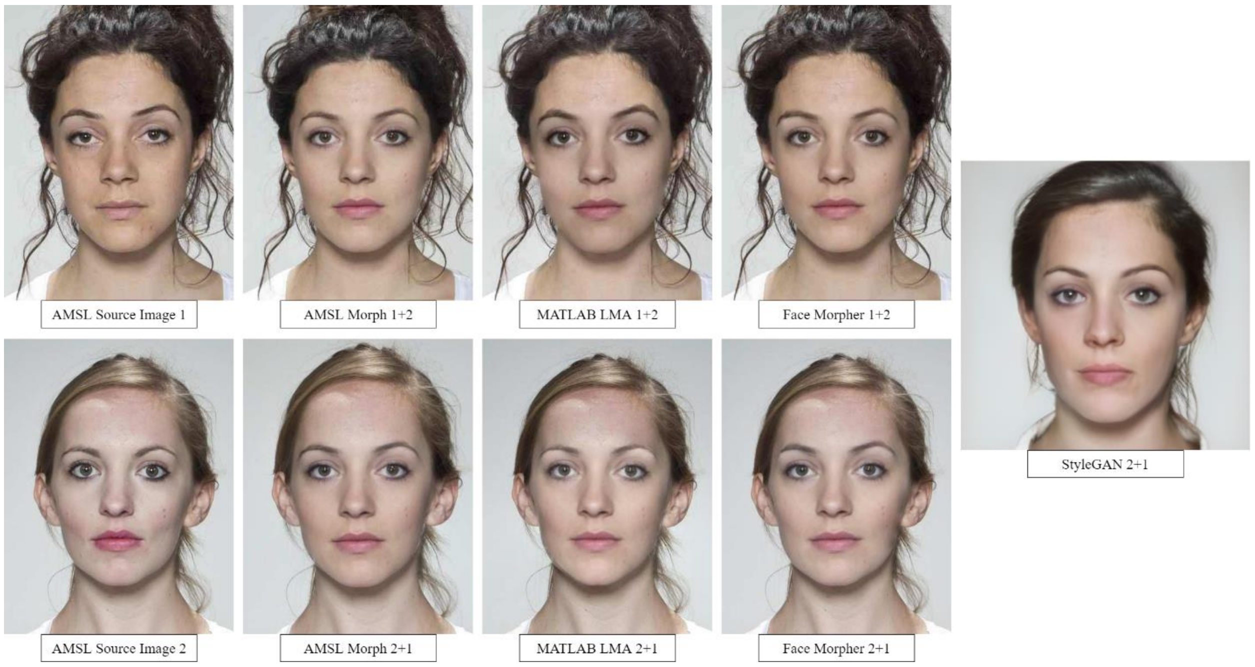

Figure 3.5: Example morphs from each method.

As you can see in Figure 3.5, each LMA method produces similar results for most of the facial region, with some differences along the border region in which splicing (Poisson blending) occurs. While the StyleGAN method results in few visual artifacts, it could be argued that the resulting morph bears little resemblance to the two source images. This will be tested in the following section on benchmarking.

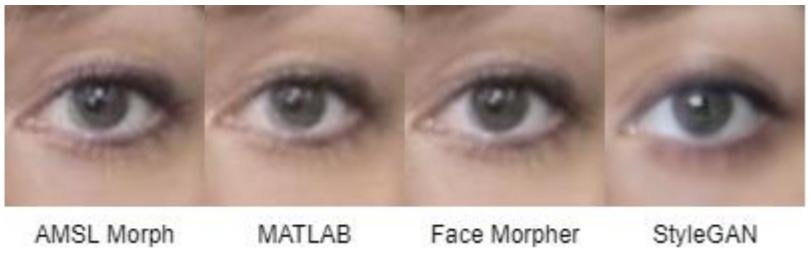

Figure 3.6: Iris artifacts.

Figure 3.6 shows iris artifacts which are commonly seen in LMA's. As you can see, Face Morpher appears to have less noticeable artifacts in this region compared to the other LMA methods. These "ghosting artifacts" are likely caused by averaging the image in a misaligned region. As expected StyleGAN2 does not experience these artifacts since it does not rely on the alignment of landmarks in the images.

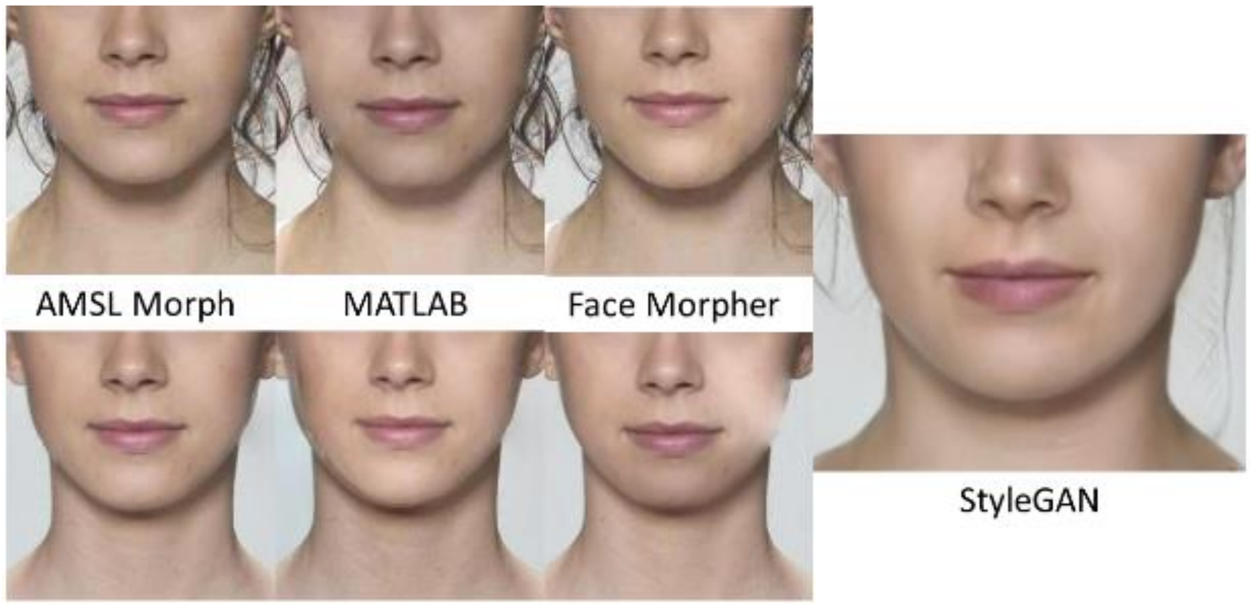

Figure 3.7: Border region artifacts.

Figure 3.7 shows artifacts around the border region of the face. We observe that these artifacts commonly occur around the cheek or chin areas of the face. In the above figure, our MATLAB implementation suffers from the same ghosting artifacts on the cheek of the image as seen in Figure 3.6. Again, this is likely caused by errors in alignment. Our MATLAB implementation can suffer from this because it doesn't directly retrieve the iris landmarks, and instead estimates where they are from the mean of other landmarks around the eyes. This can cause error in knowing the location of the irises and thus cause error when scaling and aligning the images based on these landmarks. The Face Morpher result suffers from an artifact we observe after Poisson image editing. This occurs when the mask for the region to Poisson edit is not perfectly cut around the face. As a result, some of the background is blended into the face region, causing these blending artifacts. We also observe that this type of artifact is more present when placing a morph onto a face which is narrower than the other contributor.

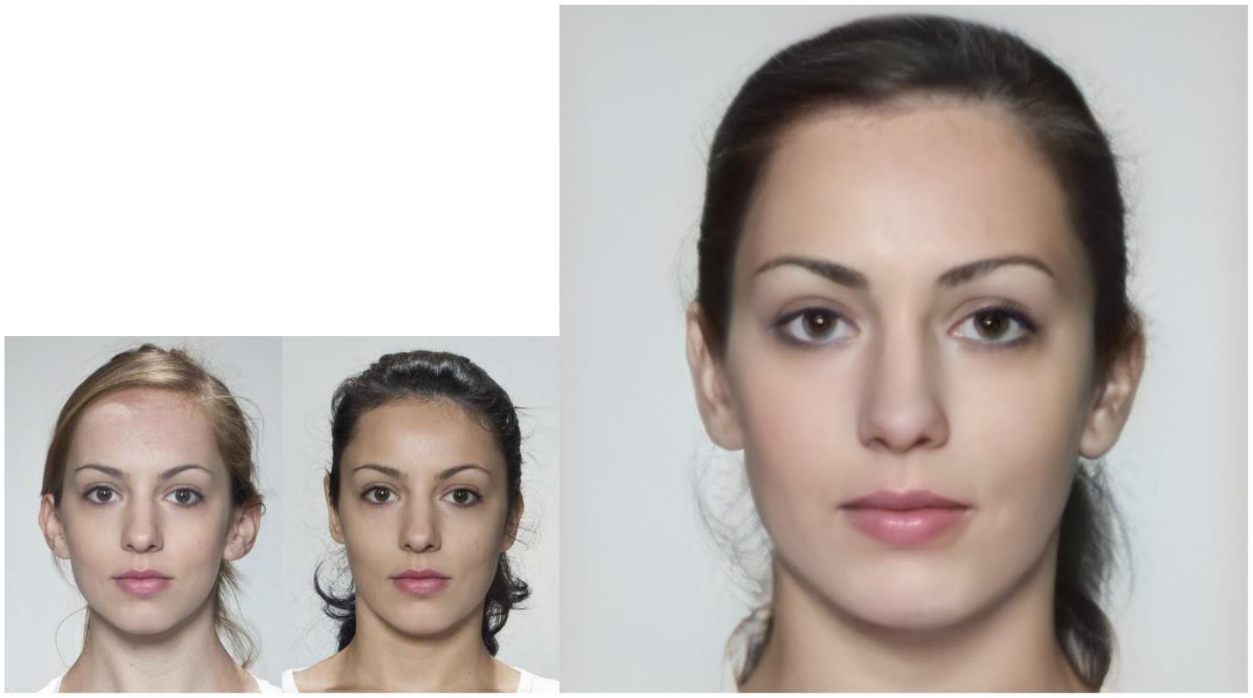

Figure 3.8: Blurriness artifacts from StyleGAN2 (right) compared to AMSL Morph (Left).

After generating the StyleGAN2 morphs, we noticed that they seemed to have a global blurriness effect (see Figure 3.8). We suspect this is caused by the alignment or embedding of smaller than optimal images. To mitigate this effect, we pass the morphs through an unsharp masking process. Through trial and error, we found that the following parameter values work well, but are likely non-optimal. The radius, or standard deviation, of the gaussian kernel was set to 2; amount, or strength, of sharpening was set to 1.5; and the threshold, or minimum contrast, for a pixel to be considered an edge was set to 0. The results can be seen below in Figure 3.9.

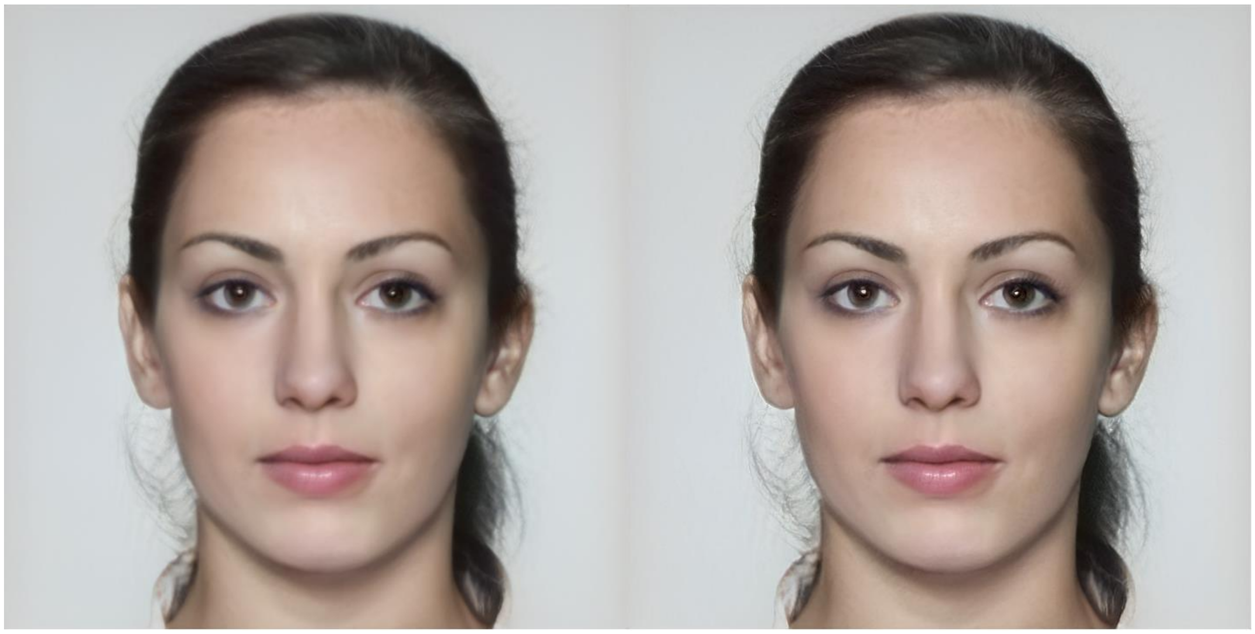

Figure 3.9: StyleGAN2 Morph (left) vs Sharpened StyleGAN2 Morph (right)

3.4.2 Benchmarking Results

To benchmark the biometric quality of these methods; we compute the MAR and rMAR of each set of morphs tested against VGG-Face using the process explained in section 3.3.2. As mentioned earlier, we choose τ such that the FAR of the system is locked at 0.1%. The summary of the results can be seen in Table 3.2. We interpret the MAR to be the rate at which morphs are recognized as the person in their source images, and rMAR as the rate a morph is recognized as the same person in all photos of the source identity.

Partition	Same Sex and Ethnicity			All		
	τ	MAR	rMAR	τ	MAR	rMAR
AMSL Morph	0.2665	**0.6929**	**0.2271**	0.2665	**0.6929**	**0.2271**
MATLAB LMA	0.2666	0.6044	*0.1206*	0.2669	0.4859	*0.0557*
Face Morpher	0.2667	0.6467	0.1759	0.2669	0.4966	0.0836
StyleGAN2	0.2665	*0.5474*	0.1526	0.2666	*0.3265*	0.0652
Sharp SG2	0.2665	0.5526	0.1561	0.2666	0.3339	0.0670

Table 3.2: Summary of Benchmarking Results. The FRR for the FRS was determined to be 0.

Note that τ is recalculated for each experiment to achieve the desired FAR. Recall that we normalize the pairwise distances between each feature vector such that they range from zero to one. Using different morph datasets can result in a different maximum distance, and thus a different normalization factor. This causes the normalized distances between "faces" to change, and thus changes the threshold for an FAR of 0.1%. However, this factor changes very little based on the morph dataset used for attacking.

To compare the datasets on an even playing field, we only test morphs that are "in common" with all datasets. Recall that AMSL restricts morphing to only those who have the same sex and ethnicity, and thus has the fewest number of morphs. Therefore, when testing the biometric quality of morphs, we only test morphs that have contributors of the same sex and ethnicity. It should be noted that for this case, half as many StyleGAN2 morphs are tested as the other datasets, due to only having one morph per pair of contributing images instead of two.

The results show that AMSL Morph has the best biometric quality, meaning that its morphs are most often recognized as the source identities, with a MAR of 69% and a rMAR of 23%. In terms of MAR, the other two LMA methods follow with our MATLAB implementation yielding 60% MAR, and Face Morpher does slightly better with 65% MAR. StyleGAN2 does the worst with 55% MAR; implying that although StyleGAN produces morphs with few visual artifacts, it has a hard time preserving biometric features. In terms of rMAR, Face Morpher does second best again with 18%. StyleGAN2 ends up with 15% rMAR, and 16% rMAR after sharpening, and does better than our MATLAB implementation which has 12%. We interpret this to mean that although our StyleGAN2 morphs are less recognizable as their contributing images, it's relatively more robust in terms of its recognizability across images as a contributing identity.

For completeness, we also test all morphs in each dataset against our FRS. These results are shown under the "All" heading in Table 3.2. As expected, these sets of morphs perform far worse than just using the "same sex and ethnicity" morphs. This shows that biometric quality of a morph suffers when morphing members of the opposite sex or different ethnicities using these methods.

3.5 Summary

In this chapter we played the role of an attacker attempting to fool an automated FRS using morphing attacks. The goal of an attacker is to create a morph that will be recognized as two different people by an FRS. We applied three methods for constructing face morphs to the AMSL Morph dataset. These methods include an open source implementation of landmark based morphing, our MATLAB implementation of [2]'s LMA combined morphing method, and an implementation of StyleGAN2 based morphing. Additionally, we use the AMSL morphs as a baseline to compare the other methods. Each method's results were compared in terms of their visual artifacts and their ability to fool an FRS. The morph sets produced for this work each provide insight into what attackers of different experience are capable of and the strengths and weaknesses of LMA and GAN based attacks.

Chapter 4
Conclusions and Future Work

The following sections contain the conclusions drawn from this work and discussions of the future work for real vs fake faces.

4.1 Conclusions

We conclude that facial action units (FAUs) can work as the sole features to base the classification of deepfake videos. We base this on our model achieving an AUC over 0.90 when detecting face morphs, and about 0.94 when further optimized. Additionally, we examined which features were most useful to a classifier for detecting deepfakes of a particular person of interest (POI). We speculate that blinking (AU45) artifacts are still one of the most useful features for the detection of deepfakes based on their prevalence in our results.

Our work on face morph attack generation and benchmarking showed that StyleGAN2 based methods produce morphs with few of the visual artifacts seen in LMAs, but have difficulty preserving biometric features. We also conclude that the biometric quality of a morph suffers when morphing members of the opposite sex or different ethnicities regardless of the choice of LMA or GAN based morphing attacks.

Due to the popularity of deepfakes, it is foreseeable that GAN based methods (in general) will improve. Thus, leading to better GAN based morphing methods. Therefore, deepfake detection methods, which make use of GAN based artifacts, would have more applicability to the world of morphing attack detection (MAD). Additionally, we propose that an effective method of combining deepfakes and face morphing would not be unlikely, as both processes attempt to swap the faces of an individual with another face. However, because we’ve shown that GAN based morphs do not preserve biometric features well, we anticipate that combining these methods would best be used for improving the forensic and visual quality of morphs.

4.2 Future Work

As mentioned in section 2.4.1 FAU correlations (features) will be ranked highly for one of two reasons, either the feature is difficult for an impostor to mimic or deepfakes naturally alter these features. While both are useful for detecting deepfakes, the former reason would be POI specific and would not help detect deepfakes of other POIs. We suspect that conducting similar experiments for multiple POIs would allow one to derive which of these features fall into the POI specific category and which are changed by deepfakes directly.

This work benchmarked the MAR and rMAR of morphing methods using VGG-Face as an FRS. MAR and rMAR varies with the similarity threshold for matching (τ). However, the dataset used for determining the τ consisted of only 102 unique identities, and τ corresponding to 0.1% FAR would likely change for a different dataset. While we feel this was a good first step, the dataset should be expanded to get a more robust measurement for the desired FAR. Additionally, we visually compared the visual and forensic quality of morphs. It should be noted that this is a subjective process, and a more quantitative approach should be done to reliably compare the visual and forensic quality of morphing methods.

We propose applying deepfakes as a splicing tool for the generation of face morphs. As we've seen, the state of the art in GAN generated imagery leaves behind few visible tampering artifacts but doesn't always preserve biometric quality. LMA's have the opposite problem, with artifacts often being introduced due to errors in alignment. Therefore, we propose to use LMA's to produce face morphs, and deepfake networks for postprocessing and splicing. To the best of our knowledge, this is a novel idea that has not yet been explored.

References

[1] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting World Leaders Against Deep Fakes.,” in *CVPR Workshops*, 2019, pp. 38–45.

[2] T. Neubert, A. Makrushin, M. Hildebrandt, C. Kraetzer, and J. Dittmann, “Extended StirTrace benchmarking of biometric and forensic qualities of morphed face images,” *IET Biom.*, vol. 7, no. 4, pp. 325–332, Jul. 2018, doi: 10.1049/iet-bmt.2017.0147.

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” in *Procedings of the British Machine Vision Conference 2015*, Swansea, 2015, p. 41.1-41.12, doi: 10.5244/C.29.41.

[4] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection,” *ArXiv200100179 Cs*, Jun. 2020, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/2001.00179.

[5] P. Korshunov and S. Marcel, “DeepFakes: a New Threat to Face Recognition? Assessment and Detection,” *ArXiv181208685 Cs*, Dec. 2018, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1812.08685.

[6] D. Guera and E. J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, Nov. 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639163.

[7] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” Jun. 2019.

[8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” *ArXiv191204958 Cs Eess Stat*, Mar. 2020, Accessed: Apr. 09, 2021. [Online]. Available: http://arxiv.org/abs/1912.04958.

[9] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, “StyleGAN2 Distillation for Feed-forward Image Manipulation,” *ArXiv200303581 Cs*, Oct. 2020, Accessed: Nov. 24, 2020. [Online]. Available: http://arxiv.org/abs/2003.03581.

[10] Y. Li and S. Lyu, “Exposing DeepFake Videos By Detecting Face Warping Artifacts,” *ArXiv181100656 Cs*, May 2019, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1811.00656.

[11] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, “Unmasking DeepFakes with simple Features,” *ArXiv191100686 Cs Stat*, Mar. 2020, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1911.00686.

[12] S. McCloskey and M. Albright, “Detecting GAN-generated Imagery using Color Cues,” *ArXiv181208247 Cs*, Dec. 2018, Accessed: Mar. 19, 2021. [Online]. Available: http://arxiv.org/abs/1812.08247.

[13] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3204–3213, doi: 10.1109/CVPR42600.2020.00327.

[14] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: a Compact Facial Video Forgery Detection Network,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Hong Kong, Hong Kong, Dec. 2018, pp. 1–7, doi: 10.1109/WIFS.2018.8630761.

[15] 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)*, Seoul, Korea (South), Oct. 2019, pp. 4431–4440, doi: 10.1109/ICCV.2019.00453.

[16] Y. Li, M.-C. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Hong Kong, Hong Kong, Dec. 2018, pp. 1–7, doi: 10.1109/WIFS.2018.8630787.

[17] P. Eckman and W. Friesen, “Manual for the facial action coding system,” *Consult. Psych Press Palo Alto*, 1977.

[18] T. Baltrusaitis, M. Mahmoud, and P. Robinson, “Cross-dataset learning and person-specific normalisation for automatic Action Unit detection,” in *2015 11th IEEE International*

Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, May 2015, pp. 1–6, doi: 10.1109/FG.2015.7284869.

[19] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, Xi'an, May 2018, pp. 59–66, doi: 10.1109/FG.2018.00019.

[20] M. Ferrara, A. Franco, and D. Maltoni, "The magic passport," in *IEEE International Joint Conference on Biometrics*, Clearwater, FL, USA, Sep. 2014, pp. 1–7, doi: 10.1109/BTAS.2014.6996240.

[21] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face Recognition Systems Under Morphing Attacks: A Survey," *IEEE Access*, vol. 7, pp. 23012–23026, 2019, doi: 10.1109/ACCESS.2019.2899367.

[22] M. Hildebrandt, T. Neubert, A. Makrushin, and J. Dittmann, "Benchmarking face morphing forgery detection: Application of stirtrace for impact simulation of different processing steps," in *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, Coventry, United Kingdom, Apr. 2017, pp. 1–6, doi: 10.1109/IWBF.2017.7935087.

[23] A. Makrushin, T. Neubert, and J. Dittmann, "Automatic Generation and Detection of Visually Faultless Facial Morphs:," in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Porto, Portugal, 2017, pp. 39–50, doi: 10.5220/0006131100390050.

[24] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.

[25] 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," p. 10.

[26] 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)*, Sep. 2019, pp. 1–5.

[27] M. Ferrara, A. Franco, and D. Maltoni, “Face Demorphing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 4, pp. 1008–1017, Apr. 2018, doi: 10.1109/TIFS.2017.2777340.

[28] 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)*, Porto, Portugal, Apr. 2020, pp. 1–6, doi: 10.1109/IWBF49977.2020.9107970.

[29] L. DeBruine and B. Jones, *Face Research Lab London Set*. figshare, 2017.

[30] A. Wolf, “ICAO: Portrait Quality (Reference Facial Images for MRTD),” 2016.

[31] K. Raja *et al.*, “Morphing Attack Detection -- Database, Evaluation Platform and Benchmarking,” *ArXiv200606458 Cs*, Sep. 2020, Accessed: Feb. 18, 2021. [Online]. Available: http://arxiv.org/abs/2006.06458.

[32] M. Tanaka, R. Kamio, and M. Okutomi, “Seamless image cloning by a closed form solution of a modified Poisson problem,” in *SIGGRAPH Asia 2012 Posters on - SA ’12*, Singapore, Singapore, 2012, p. 1, doi: 10.1145/2407156.2407173.

[33] A. Rottcher, U. Scherhag, and C. Busch, “Finding the Suitable Doppelga¨nger for a Face Morphing Attack,” p. 7.

[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” 2018.

[35] M. Ferrara, A. Franco, and D. Maltoni, “On the effects of image alterations on face recognition accuracy,” in *Face recognition across the imaging spectrum*, Springer, 2016, pp. 195–222.