Detection of Face Morph Attacks on Facial ID Verification Systems

M. Cem Gülümser - 22003430
Department of Computer Engineering
Bilkent University
cem.gulumser@ug.bilkent.edu.tr

Instructor: Shervin Rahimzadeh Arashloo
Department of Computer Engineering
Bilkent University
s.rahimzadeh@cs.bilkent.edu.tr

Abstract:
Biometric identity verification is employed widely around the world by major companies and governments to ensure social security. With the help of modern face recognition technology, it is much more convenient to confirm the identity of people. In recent years, however, criminally minded individuals devised a technique to hide their identity without being detected by face recognition. This technique is called "face morphing", where the facial image of two or more people are combined so that the resulting image looks similar to both subjects. This causes the exploitation of certain governmental resources by wrong individuals, illicit international migration and invasion of privacy. In this work, the vulnerability of face recognition systems are analyzed, and several face morphing algorithms are compared for their likelihood of detection.

1. Introduction

An important part of this study focuses on the process of pattern classification. Being the study of extracting raw data from a model and distinguishing it from other models depending on its features, the working mechanism of modern pattern classification systems is the main target of face morph attacks which will be further explained in this chapter. The groundwork regarding this field is therefore essential to be noted.

a. Pattern Classification Systems

A modern pattern classification system (PCS) works in a few different steps to process and categorize its input. This categorization is then delivered as a simple output indicating the category of the input, or sometimes a decision based on the determined category. Classifiers only work with their specified models, i.e, inputs that generate features. In order to make a decision, the classifier has to sense the input, separate it from irrelevant noise, extract the features of the model and categorize it based on the features. This process is divided into five steps: (i) *Sensing,* (ii) *Segmentation and Grouping,* (iii) *Feature Extraction,* (iv) *Classification and* (v) *Post-processing* [1].

As a subclass of pattern recognition systems, face recognition systems (FRSs) execute these steps in the process of detecting and matching faces. The system initially receives a raw input provided by the sensor. This raw input has to be preprocessed in order for it to be properly recognized by the PCS. Isolation of the

model from others makes it easier for the system to distinguish its beginning and end. Otherwise, overlapping models may cause incorrect segmentation and thus inaccurate classification. After an isolated model is provided to the system, its features are to be extracted and passed to the classifier. The problem here is that these features have to be distinguishing between categories. For example, the length of a tree might not be enough to distinguish it from other species, but the shape of its leaves might be a good feature to discriminate it from other trees, or even tell its species. Feature extraction defines a model with the probability density of its features which are statistically similar for the objects of the same category, and distinct with respect to other categories. The selection of features for a PCS is up to the designer, but these features ultimately determine the success ratio of the classifier. Another problem that might arise in feature selection is their variation depending on reasons unrelated to the category of the model. Noise, translation, rotation, scaling and corruption are some of the main reasons that may cause a shift in features. For instance, the post-processing (the concept of post-processing will be explained in detail in the following chapters) of biometric mugshots may cause the resolution to change, resulting in varying classifications [2].

The role of classification in the PCS is to receive the features provided by the extractor and represent them as feature vectors in the *i* dimensional Euclidean space.

$$x = \left(x_1, x_2, x_3, \ldots, x_i\right) \in R^i$$

Where x denotes a feature vector of a given model, and its components representing the singular features assigned to that model. After proper training, the classifier determines a *decision boundary* to categorize the model according to its feature vector. The main problem with classification is the presence of noise, which is defined as "any property of the sensed pattern which is not due to the true underlying model but instead to randomness in the world or the sensor." [2] Noise causes feature values of objects to be inaccurate, making the classification more difficult. Finally, the output of the classifier is received by the post-processor, which then recommends a specific action. However, every action has a corresponding cost, error rate and context. The purpose of the post-processor is to suggest an action that would minimize the expected cost as well as the risk of the action.

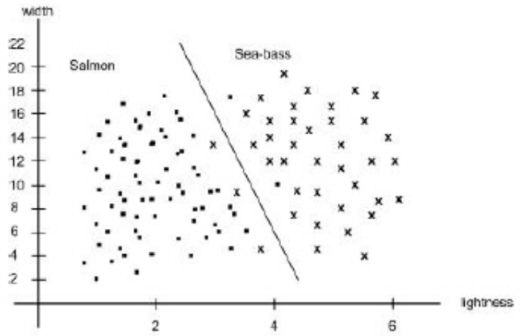


Figure 1: A linear decision boundary of a PCS is drawn. This case represents a classification between Salmon and Sea-bass according to absolute width and lightness of the fish [10].

2. Related Work

Previous work on the subject suggests the intricacy of the matter, as there are numerous factors affecting the detectability of a face morph attack. The effectiveness of the FRS and its pattern classifier, algorithm of the face morph attack, and the efficiency of the face morph detector all affect the success ratio of detection.

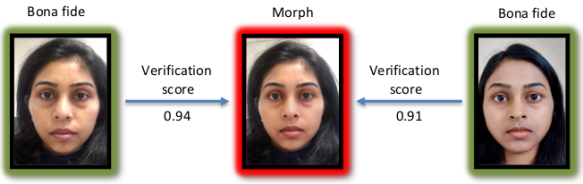


Figure 2: A morphed face of two individuals [11]

a. Face Morph Attacks (FMAs)

There are numerous ways to deceive or corrupt a FRS. One of such methods is presenting an image of the combined face of two or more individuals to a FRS which is known as a FMA. Face morphing is a way to fuse multiple biometric images in one image, which resembles both faces. Figure 2 represents an example of a FMA that can possibly be used against a facial ID verification system (FIVS) [11].The morphed image presented above has a conspicuously high verification score against both subjects. This means that if this image is compared with a database of other facial images containing the face of either of the subjects, it would be verified against one of the subjects, and would result in a serious security concern. Such instances of FMAs makes governments and private companies question the reliability of facial identification as a security tool [9].

There are many proprietary or open-source face morph algorithms (FMALs) that could be used to conduct a FMA. Some of these softwares are Abrosoft FantaMorph, FaceMorpher, OpenCV, Magic Morph, FaceFusion, Face Swap Online, UBO-Morpher, etc. Most of these softwares or algorithms leave some artefacts that can be detected through different methods. Since it is unknown what type of algorithms criminals use, it is safer to test different algorithms and benchmark the detectability

Figure 3: Example of a morphed face of two individuals with different algorithms. From left to right: Subject 1, FaceFusion, FaceMorpher, OpenCV, UBO Morph, and subject 2 [9].

of each. Figure 3 illustrates the usage of different FMALs with the same subjects. Inspecting the morphs, it is evident that there are some artefacts due to the distinction of two faces. The first morph was created with the proprietary *FaceFusion* morphing algorithm [3]. The source code is not available for this algorithm, but the artefacts are effectively hidden by blending two images. The next morph is created with an open-source Python based face morphing algorithm named *FaceMorpher* that employs the STASM wrapper to find features on the face [8]. Using these feature points, triangles are formed and distorted in the morphing process. There are evident artefacts in the resulting image. The third FMAL is named *OpenCV* that employs Dlib for feature and landmark detection [4] and is similar to the previous FMAL. This one creates the feature points at the edge of the face and then distorts them in the morphing procedure. Similar artefacts are observed in the morphed face. The last FMAL is *UBO Morph* which is designed at the University of Bologna. This algorithm receives two inputs, the faces of the subjects and the landmark points on each face. Similar to aforementioned FMALs this morph is carried out by triangulation, averaging and blending. Again, artefacts on the forehead, hair and eyes are observed. It is important to note that the faces in this example are clearly different from each other which makes the morph more difficult. In a real life scenario, FMAs employ subjects that look alike, where the morphed image is very similar to both (see Figure 2) [9].

In general, FMALs can be placed under two categories in terms of their working mechanism. These categories are *Landmark Based Morph Generation* (LBMG) and *Deep-Learning Based Morph Generation* (DLBMG). In the first category of LBMG, the algorithm first attempts to find the landmark points on both faces, such as the eyes, nose, mouth and the edges of the face. Next, these landmark points are warped into a single image to generate the morph [11]. There are many ways to achieve the combination of landmark points. A common method of morphing is using Delaunay Triangulation, where the pixels of both images are adjusted to create triangles [6]. Other methodologies employ image partitioning, where feature points in the face are extracted and then morphed into one with bilinear and affine transformations [7]. Other algorithms include using the Bayesian Framework to calculate the maximum probability morph field to create an optimal morph between two images [5]. Finally, these FMALs use blending factors to morph the images more similar to one subject or another. A blending factor of 0.5 is equally similar to both images, and is called a *symmetric morph* [11].

The second type of FMALs is DLBMGs, which employs the deep-learning technology to generate morphed images. The development of Generative Adversarial Network (GAN) has laid the groundwork for the development of such algorithms. Some GAN based methods include MorGAN, StarGAN and StyleGAN. MorGAN face morphing architecture employs a specifically trained neural network where the two given faces are encoded into the latent space, linearly interpolated with a factor of 0.5 and is decoded into the image

space [13]. Whereas MorGAN can generate images with 64 x 64 pixels, StyleGAN has increased this dimension to 1024 x 1024 with the use of similar technology. Other deep learning based methods include MIPGAN-I and MIPGAN-II, which demonstrate morphs with much higher qualities compared to landmark based methods. The artefacts of morphed images are minimized and hidden with sophisticated methods, and to distinguish between a morphed image and a bona fide image is even difficult for humans [11].

b. Morph Attack Detection (MAD)

As a defense against such attacks, various Morph Attack Detection (MAD) algorithms have been created, some of them having considerably high success ratios. MADs fall under two categories in terms of their requirement of a reference image of a live subject. *No-reference* MADs do not require an additional input other than a morphed or a bona fide image. It is simply a pattern classification system with two states of nature (i.e categories) ω_1 for the state that the given image is morphed, and ω_2 for the state that the image is bona fide. On the other hand, *differential* MADs expect two images as an input and decide whether the potentially morphed image actually belongs to the live subject or not. This is achieved through the comparison of two images and outputting three states of natures: The image is real and belongs to the subject, the image is a morphed image of the subject, the image is a morphed image but the subject has no contribution to the morph [9].

The main topic of this paper is no reference MADs. As stated before, such MADs have two states of nature in their decision set:

$$D = \{\omega_m, \omega_b\}$$

Where D is a finite set of two states of nature for a morphed state and a bona fide state. There are several methods of detecting a face morph on an image. Figure 4 illustrates most common MAD methods. The scope of this research covers only the MADs that utilize *texture descriptors* and *digital forensics.* Texture descriptor based MADs (TDBMADs) attempt to detect the artefacts that are created after the morph

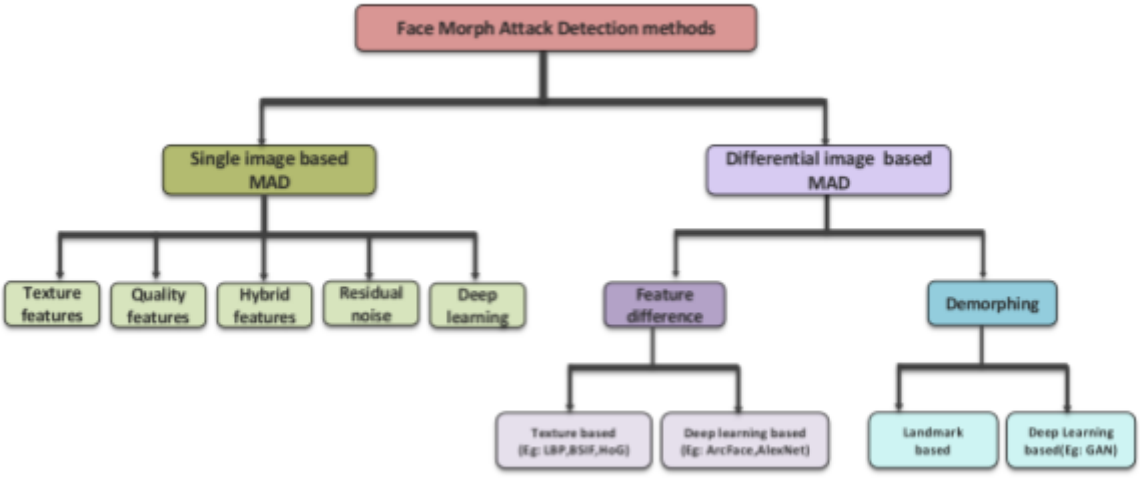


Figure 4: Table of most common face morph attack detection algorithms. [11]

process to assign a category to the input. Distortions, smoothed edges, ghost and half-shade remnants are some of the common artefacts that can be seen in morphed images (see Figure 3). Texture descriptors such as local binary patterns (LBP), binarised statistical image features (BSIFs) or weighted local magnitude patterns (WLMPs) are some of the common methods used in TDBMADs [9]. In other words, texture features can be passed into the pattern classifier of a MAD to decide if an image is morphed.

c. Effects of Post-Processing

Digital forensics refers to MADs classifying images based on quality and noise. Quality can change as a result of post-processing of the image and noise is visible on the image as a result of the randomness of the world and the dysfunction of the equipment employed in the process of sensing the input. Forensics-based MAD (FBMAD) assigns features based image quality to the feature vector. Assuming that a feature vector x is extracted from an image that was given as an input image to a FBMAD, it can be denoted as follows:

$$T = \{t_1, t_2, t_3, \ldots, t_i\}$$
$$Q = \{q_1, q_2, q_3, \ldots, q_j\}$$
$$N = \{n_1, n_2, n_3, \ldots, n_k\}$$

$$x = (T, Q, N)$$

Where T denotes the set of texture features, Q denotes the set of quality features and N denotes the noise features. A FBMAD utilizes all three of these features that arise as a result of post-processing to classify its input. So the assigned feature vector contains all of the extracted features as a result of post-processing [9].

In photography, post-processing refers to the application of effects that manipulate the color, contrast, sharpness or brightness of the image. In the scope of this paper, post-processing is defined more generally as any change or alteration in the image after the morph process. This can include compression, resizing, printing, scanning. Figure 5 illustrates a morphed image of two subjects which was post-processed by printing and scanning. The difference in colors and shading between the images are evident. In the case of editing the image as a way of post-processing, the image is intentionally manipulated to deceive a FRS. Deriving from this, it is important to note that post-processing makes MAD more

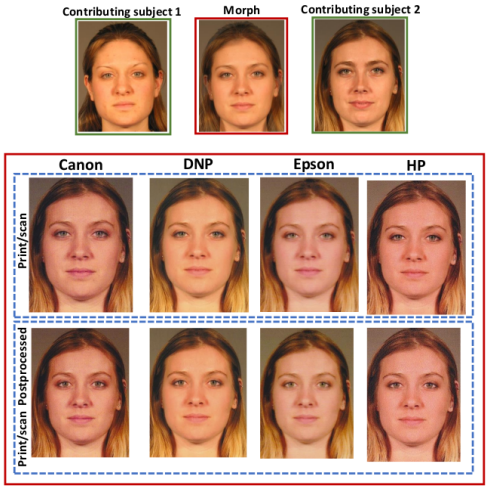


Figure 5: Effects of various printing and scanning on a morphed image [11].

inconvenient as it can be used to hide and erase artefacts as a result of the FMA. Especially in FBMAD where the extracted features depend on the quality and noise of the image, classification becomes much more difficult. So, how does a MAD algorithm overcome such difficulties? What is the crucial key to design a MAD system that can see past the effects of post-processing and make a risk-optimized classification?

3. Evaluation of Common Face Morph Detection Techniques

One thing to keep in mind is that a FRS system never works alone, meaning that in border security or in social surveillance, FRS systems compare their results against a *database* of bona fide images. The pattern classifier of a FRS system presents *comparison scores* for each image in the database, and passes this as a parameter to its post-processor. The post-processor evaluates the *risk*, and returns a classification for the image, which is the person that the face belongs to. A significant derivation from this is that for a FMA to pose a considerable threat to a FRS, the morphed image has to be verified against at least one person. This person may be the actual owner of the face, or not. So how do we decide whether to put an image to a test against a MAD system? In the next sections of this chapter, first the databases that will be used in the experiment will be outlined (a). Second, the FRS employed in the experiment will be introduced in detail (b). Next, the results of the conducted experiment with respect to different FMA will be presented (c) , and their respective threat analysis will be made (d). Finally, the highest threat posing morphs will be evaluated in scenarios if they were to be presented against different MAD algorithms (e). This will reveal which MAD algorithm would be the most successful in detecting the highest risk posing morphs.

a. Database

The database used in this research was entitled “AMSL Face Morph Image Data Set” [14] where the morphed faces of 200 people with 5 different FMAL are included. The original faces are retrieved from another database named “Face Research Lab London Set" [16]. The morphed face image data set is generated by the selected bona fide images and the morphing was accomplished through a process reported by Neubert et. al [15]. This database contains morphed faces of people from different ages and ethnicities. There are morphs present from 5 different face morph algorithms: Webmorph, StyleGAN morph, OpenCV, facemorpher and AMSL. The basic working mechanism of these algorithms were explained in chapter 2.a. In addition, the morphed faces generated in this data set are biometric images of the same quality and size, so these factors are constant during the experiment.

b. Face Recognition Algorithm

The face recognition system of this research was created using the pre-trained neural network GoogleNet on MATLAB. As this network had been trained to classify daily life objects, it had to be modified for face recognition [17].

(i) Preparing the Dataset:

The dataset was created with the original faces of 10 different people imported from the Face Research Lab London database. For each person, 6 different images of their faces were included in the dataset. So in total, the dataset contained 60 images. After the preparation of the dataset, each of the 6 images belonging to the same subject were categorized under a folder entitled with the number assigned to that subject (for example the images belonging to the 6th person were in the folder named *'Subject6'*). This way, the classifier could label the input image with the corresponding folder name [17].

(ii) Layer Modification:

In order for the network to make proper classifications of faces, the classifier and the output layers had to be altered. To achieve this, a transfer learning method was used in which the task specific layers of the network were switched with new layers that can recognize and classify faces. In particular, the 142nd (loss3-classifier) layer which is responsible for learning the extracted features and the 144th (output) layer which classifies the outputs (see Appendix 1 for the graphical analysis of the last layers). The 142nd layer was replaced with a facial feature learner and the final layer was replaced with a face classifier [17].

(iii) Preprocessing & Augmentation:

An essential part of face recognition is the preprocessing of the input images before feature extraction. This is the cropping of the image where only the face of the individual is left on the image. This was manually done before the preparation of the dataset. Furthermore, the network only accepts images with particular dimensions, which is 224x224 pixels. So the input images had to be resized before being passed into the neural network for classification. Another important point to consider is the image augmentation which is the practice of expanding the dataset to allow training with more diverse data. To achieve this, the "imageDataAugmenter()" function was employed. Images were randomly translated, scaled, reflected and modified to allow diversification [17].

(iv) Training:

The aforementioned network is already trained, however, the replaced layers still require training to make proper classifications of faces. Training of these layers was done with the "trainNetwork()" function. From the input dataset, 4 of the images were used for training and 2 of the images were used for validation at each epoch of training. For better accuracy, the dataset was shuffled after the completion of each epoch. The training options of the network can be found in Appendix 2 and the details of the training process can be found in Appendix 3 [17].

(v) Verification Scores:

After training, the network was able to make proper classifications of faces. The accuracy of the classifications was given to be %100 (see Appendix 3) which means that the FRS can classify all of the validation dataset correctly. However, this result should not be interpreted as a perfect FRS, because the training dataset was small and it was quite

probable for the FRS to classify all of them correctly. After the training process, the FRS was tested manually for the verification scores of the individuals that were present in the dataset. For clarity, a verification score is the maximum probability output of the FRS which belongs to the classification category provided by the FRS. Table 1 shows the verification scores for the bona fide images of the subjects included in the dataset. For more precise results, the dataset was trained 5 different times and the mean value of the verification scores was calculated. The verification scores given in the table below are the mean values, see Appendix 4 for the actual verification scores for each trial. The fluctuations in the verification scores of each subject can be explained by the similarity of other subjects to that particular one. For example, for the first subject, the probability of the output classification to belong to that subject is quite high because that subject has distinct facial features with respect to others. On the other hand, subjects 3 and 4 look alike and the probability of classification decreases since the input image might belong to the other subject.

Subject	***Verification Score***
Subject 1	0.998
Subject 2	0.965
Subject 3	0.775
Subject 4	0.658
Subject 5	0.555
Subject 6	0.778
Subject 7	0.702
Subject 8	0.796
Subject 9	0.993
Subject 10	0.888

Table 1: The corresponding mean verification scores for each subject present in the dataset.

c. Experiment with Different FMA Algorithms

In this section, the results acquired from the benchmarked face morph attack algorithms will be presented and analyzed. However, it is first important to understand the rationale behind this experiment as well as its procedure to better understand the results. In a real life situation, the attacker will find a lookalike accomplice to morph his face with. The morphed face will be compared with that of the accomplice and the criminal. However, the database that the comparison will take place against will not contain both of the faces. For instance in a passport application, the face of the accomplice will not be present in the database as this is a new application for a passport. Similarly, in border security, since the criminal will attempt to enter a country which he/she is not a citizen of, the database will not have his/her face. The only comparison will be between the printed image on the passport (in this case a morphed image) and the database containing the image of the accomplice who does not have a criminal background. If the passport image is verified

against the accomplice's face in the database, the criminal will safely be allowed into the country. To model this situation, the following experiment is devised.

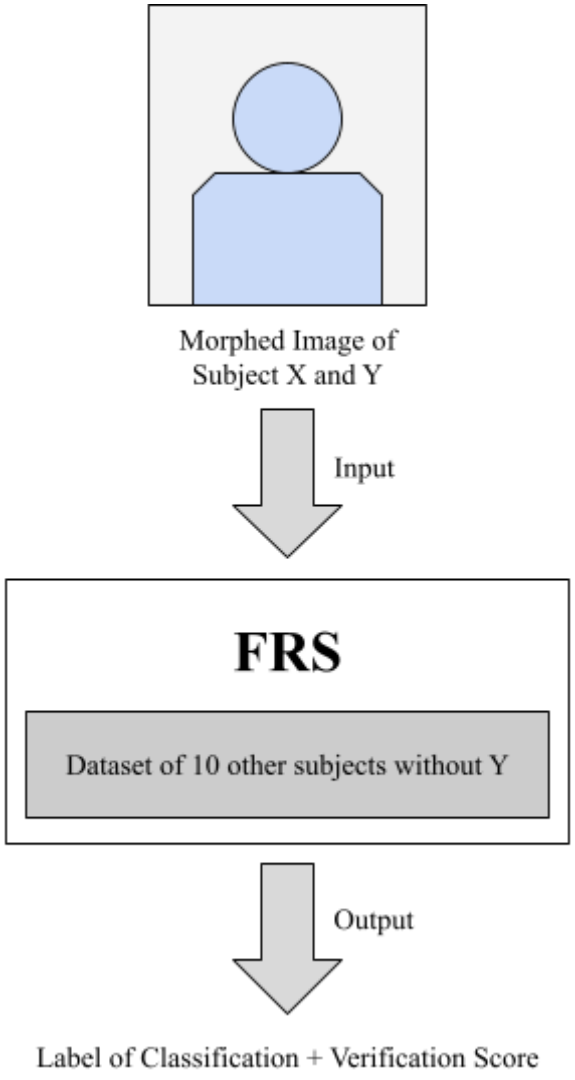


Figure 6: The model of the experiment. For a given subject X, his morph with another subject Y is given as an input to the FRS. The verification score of X for that particular morph is obtained from the FRS where Y is not present in the dataset.

First, two lookalike subjects X and Y are selected from the dataset and their corresponding morphed faces for each FMA is gathered from the database. Next, one of the subjects, let us say Y, is taken out of the dataset and an arbitrary subject is added. After training the FRS with the modified dataset, the morphed faces are provided to the FRS as input and the outputs of the FRS are recorded as the verification scores for Subject X. Next, the experiment is repeated with the same logic but this time, instead of taking out the bona fide images of Y from the database, images of X are taken out of the dataset. The network is trained again and the verification scores for Subject Y are obtained. Figure 6 is a depiction of the experiment. In this experiment, 5 morph algorithms were benchmarked:

- Landmark Based: Webmorph
- Landmark Based: AMSL
- Landmark Based: FaceMorpher
- Landmark Based: OpenCV
- Deep-Learning Based: StyleGAN

The acquired results are indicated in the figures below. Figure 7 illustrates the label of classification for each morph (Webmorph, StyleGAN, OpenCV, Facemorpher and AMSL from left to right) and the verification score associated with that label. In similar fashion Figure 8 for subjects 3 and 4, Figure 9 for subjects 5 and 7, Figure 10 for subjects 6 and 8, and Figure 11 for subjects 9 and 10. See Table 2 for the complete representation of the experimental results.

Subject 1 - Subject 2: (webmorph, stylegan, opencv, facemorpher, amsl)

Figure 7: Subject 1 (below) - Subject 2 (above)

Subject 3 - Subject 4: (webmorph, stylegan, opencv, facemorpher, amsl)

Figure 8: Subject 3 (above) - Subject 4 (below)

Subject 5 - Subject 7: (webmorph, stylegan, opencv, facemorpher, amsl)

Figure 9: Subject 5 (above) - Subject 7 (below)

Subject 6 - Subject 8: (webmorph, stylegan, opencv, facemorpher, amsl)

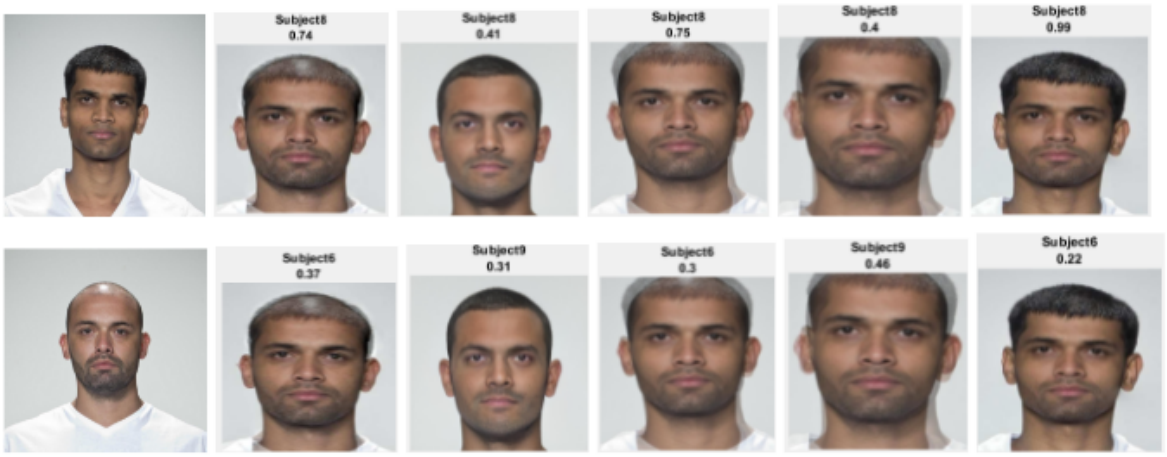


Figure 10: Subject 6 (below) - Subject 8 (above)

Subject 9 - Subject 10: (webmorph, stylegan, opencv, facemorpher, amsl)

Figure 11: Subject 9 (above) - Subject 10 (below)

Morph	*Subjects*	*Verification Score*				
		Webmorph	StyleGAN	OpenCV	Facemorpher	AMSL
1 - 2	Subject 1	0.95	0.87	0.85	0.57	1
	Subject 2	0.75	0.91	0.89	0.79	0.56
3 - 4	Subject 3	0.91	0.98	0.89	0.88	0.95
	Subject 4	0.94	0.97	0.90	0.88	0.96
5 - 7	Subject 5	0.77	0.70	0.66	0.41	0.88
	Subject 7	0.83	0.55	0.90	0.91	0.64
6 - 8	Subject 6	0.37	0.31	0.30	0.46	0.22
	Subject 8	0.74	0.41	0.75	0.40	0.99
9 - 10	Subject 9	0.99	0.25	0.89	0.9	0.97
	Subject 10	0.91	0.53	0.53	0.84	0.45

Table 2: Comparison scores for each subject for the given morphs. The cells highlighted in red indicate that the subject was not classified as the one in that row. In such cases, the similarity score is considered to be 0. The blue cells indicate the maximum morph verification scores of each subject.

d. Threat Assessment & Verification Threshold

In their assessment of morphing techniques, Scherhag et. al, proposed a metric for the evaluation of the impact of a morphing attack named Mated Morph Presentation Match Rate (MMPMR). In this instance, *mated morph comparison* refers to the comparison of the morphed image to a contributor subject. The MMPMR function denoted with F is as follows [18]:

$$F(\tau) = \frac{1}{M}. \sum_{m=1}^{M} \{\left[min(n = 1,\dots, N_m)S_m^n\right] > \tau\}$$

where τ is the verification threshold, S_m is the mated morph comparison score of the n-th subject of morph m, M is the total number of morphed images and N_m the total number of subjects contributing to morph m [18]. An example threat analysis metric of morphed images can be created with the corresponding comparison scores of subjects as given in Figure 6. In this case, the verification threshold is *subject independent* and is set to a constant value of 0.5.

In the case of the aforementioned experiment, the only unknown to calculate the MMPMR for the vulnerability assessment is the verification threshold. It is, however, one of the purposes of this research to find an optimal verification threshold based on the results so that a minimum number of morphs will pose a threat against the FRS. To achieve this, two kinds of verification thresholds will be calculated: Subject dependent and subject independent verification thresholds.

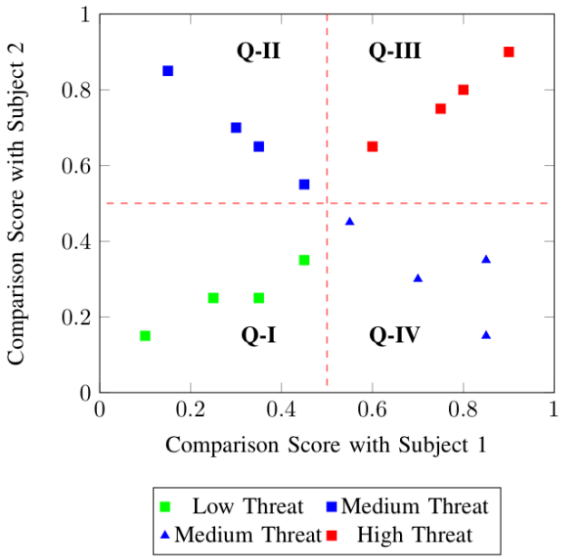


Figure 12: Threat assessment for exemplary FMAs with their given comparison scores [11].

(i) Subject Dependent Verification Threshold

When an FRS implements subject dependent verification threshold to recognize faces, each subject gets their own threshold value in order for an image to be classified as belonging to that subject. To determine this value, it is necessary to consider the verification score of the bona fide images of the subject, so that the verification threshold should be just below the minimum bona fide image verification score. If the threshold is implemented this way, the FRS is guaranteed to make no mistake while classifying bona fide images. However, if there are FMAs that can obtain verification scores above the threshold, they can easily pose a threat to the recognition system. To see the results of this methodology in the experimental results, Table 3 is prepared

with the subject dependent verification scores for each subject.

Subject	***Verification Threshold (τ)***
Subject 1	0.98
Subject 2	0.90
Subject 3	0.72
Subject 4	0.50
Subject 5	0.41
Subject 6	0.61
Subject 7	0.57
Subject 8	0.45
Subject 9	0.98
Subject 10	0.70

Table 3: The corresponding verification thresholds for each subject present in the dataset.

One limitation of this methodology is that the verification threshold is just below the minimum value, and the minimum value can be considerably low for some subjects (see Appendix 4). When this is the case, the FRS becomes prone to classifying a morphed image as one of the subjects. In other words, when the verification threshold is set dependent on subjects, subjects with lower verification scores will pose greater threat to the FRS. On the other hand, if the FRS can make more sensitive classifications, the fluctuations would be minimized and vulnerability would be decreased.

(ii) Subject Independent Verification Threshold

The idea behind setting a subject independent verification threshold is that a single verification threshold would be applied for every classification such that it would allow the minimum number of morphed images to be verified against its contributing subjects while allowing the maximum number of bona fide images to be classified as their real owner. This optimization problem could be solved with certain complex methods, but a simple way to find this threshold would be to calculate the Mean Maximum Morph Verification Score (MMMVS) from the experimental data and to find the middle value between the MMMVS and the Mean Subject Verification Score (MSVS). To elaborate, MMMVS is defined as the sum of the maximum morph verification score (see the blue highlighted cells in Table 2) of each subject outputted by the FRS divided by the number of subjects. MSVS is the mean value of the verification scores of the subjects (see Table 1). Calculating these values, it is possible to find the subject independent verification threshold as:

$$MMMVS = 0.886$$
$$MSVS = 0.811$$

$$\tau = 0.848$$

(ii) Analysis

The metric presented in part d (MMPMR) indicates that a morph presents a significant threat if all of the contributing subjects are

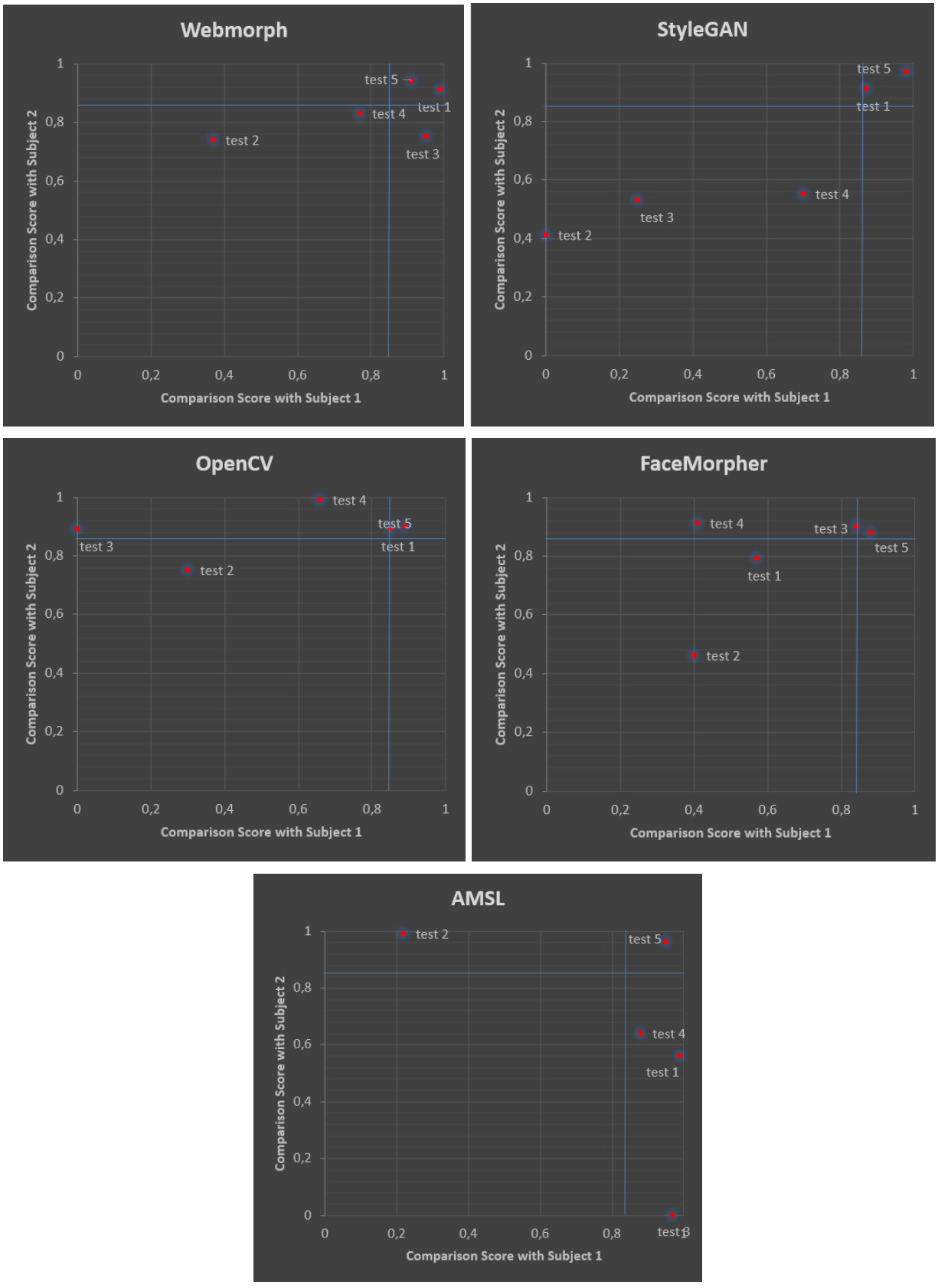
verified against it. If a given morph can be verified against both subjects, then it is said that the morph is successful. Using this idea, the same metric can be constructed for this experiment. Figure 13 depicts the experimental data in graphical form. The blue lines in each graph represent the subject independent verification threshold. Each data value has a given test number. Test 1 data point indicates the morph between Subjects 1 and 2 at every graph. Similarly, Test 2 indicates Subject 3 and 4, Test 3 indicates Subjects 5 and 7, Test 4 indicates Subjects 6 and 8, and Test 5 indicates Subjects 9 and 10.

If the verification threshold is set independently, the position of the data point with respect to the blue lines will indicate whether that particular morph is successful or not. For the Webmorph FMAL, tests 1 and 5 present a high threat to the FRS since both subjects have a verification score greater than the threshold. Test 3 has a medium threat because the second subject is not verified and falls under the verification threshold. Tests 2 and 4 pose a low threat to the FRS since none of the subjects are verified. In general, if a datapoint is located in the upper right quadrant, then it has a high threat against the FRS. Datapoints at the upper left or lower right pose a medium threat, and the lower left quadrant pose a low threat to the FRS. Judging by the graphs, it can be concluded that only 7 of the morph trials pose a significant threat against the FRS. Those morphs are the only ones that have the potential to be successful in a real life situation. Based on the definition of MMPMR, the highest threat posing face morph algorithms would be Webmorph and StyleGAN. However, morphs created with Webmorph are more prone to falling into the upper right quadrant, whereas StyleGAN morphs are more scattered around the graph. This shows that the most significant threat is caused by the Webmorph algorithm when the verification threshold is determined independent of the subjects.

Considering the subject dependent verification threshold case, graph representation is not convenient. With a simple comparison of verification threshold and morph verification scores for each subject, it is possible to determine which and how many morphs will be successful against the FRS. Table 4 is a modified version of Table 2 where the green highlighted cells indicate the morphs that have a higher verification score than the threshold for that particular subject. If both subjects contributing to the morph are green, then the morph is successful. In total, there are 9 morphs that pose a significant threat to the FRS and the highest threat posing FMAL is again Webmorph.

In conclusion, subject independent verification threshold decreases the number of high threat morphs. Implementing this threshold would be safer in a real life situation. One essential point to note is that depending on the sensitivity of the FRS and the quality of the morphs, subject dependent threshold might give better results. As it can be seen from Appendix 4, verification scores for each subject fluctuate greatly, and this makes subject dependent threshold less efficient. To better understand which one is the most effective, experiments like the one presented in this research have to be conducted.

Figure 13


Webmorph
Comparison Score with Subject 2
Comparison Score with Subject 1
test 5
test 1
test 4
test 2
test 3
StyleGAN
test 5
test 1
test 4
test 3
test 2
OpenCV
test 4
test 5
test 3
test 1
test 2
FaceMorpher
test 4
test 3
test 5
test 1
test 2
AMSL
test 2
test 5
test 4
test 1
test 3
0
0,2
0,4
0,6
0,8
1

Morph	*Subjects*	*Verification Score*				
		Webmorph	StyleGAN	OpenCV	Facemorpher	AMSL
1 - 2	Subject 1	0.95	0.87	0.85	0.57	1
	Subject 2	0.75	0.91	0.89	0.79	0.56
3 - 4	Subject 3	0.91	0.98	0.89	0.88	0.95
	Subject 4	0.94	0.97	0.90	0.88	0.96
5 - 7	Subject 5	0.77	0.70	0.66	0.41	0.88
	Subject 7	0.83	0.55	0.90	0.91	0.64
6 - 8	Subject 6	0.37	0.31	0.30	0.46	0.22
	Subject 8	0.74	0.41	0.75	0.40	0.99
9 - 10	Subject 9	0.99	0.25	0.89	0.9	0.97
	Subject 10	0.91	0.53	0.53	0.84	0.45

Table 4

e. MAD Evaluation

For security reasons, most MAD algorithms are kept undisclosed. Revealing how a MAD algorithm works might enable criminals to take precautions against their working mechanism, which would render them useless. For this reason, an experiment on this topic cannot be conducted. However, some basic information on how a MAD works are presented in the related work on this topic, which was discussed in chapter 2.b. As it was previously stated, only no-reference MADs will be evaluated in this research. Looking at the previous part of this chapter, it was established that the highest threat posing morphs were created using Webmorph and StyleGAN. So these two algorithms will be evaluated in this part.

(i) Webmorph

Webmorph works in a similar fashion to FaceFusion algorithm discussed in chapter 2.a. The algorithm first finds the feature points on the face and uses triangulation to morph the faces. The artefacts created after the morph are not hidden and easily visible in the morphs. So any TDBMAD will be sufficient to categorize these morphs as a morphed image. Venkatesh et. al propose various texture based approaches that can detect morph artefacts using Local Binary Pattern and Binary Statistical Image Features [11]. It can be concluded that the

morphs created with this algorithm will fail to be classified as bona fide images if they were to be put into a MAD algorithm.

(ii) StyleGAN

StyleGAN is a more sophisticated FMAL compared to Webmorph. It employs Generative Adversarial Network to morph faces as well as to get rid of the artefacts created after the morphing process, which makes it more difficult to be detected. Looking at the morphs created with this algorithm, it is almost impossible to distinguish it from the face of a real person, so the MADs that use texture based approaches or image degradation are useless against this FMAL. There are certain no-reference MAD algorithms that are proven to be effective against such high quality morphs. These MADs employ convolutional neural networks to classify an image as morphed or bona fide by using feature fusion of fully connected layers [11]. However, there is no guarantee that the morphs generated with this algorithm will be detected, which makes it the most dangerous.

4. Conclusion

Face morph attacks can deceive face recognition systems by getting verified as one of their contributing subjects. If all of the contributors are verified, then it is assumed that the morph is successful. To prevent FMAs from succeeding verification thresholds should be set to optimal values which would allow the least number of morphed faces to be recognized, and maximum number of bona fide faces to be classified as their original owner. The experimental results indicate that the most effective way to achieve this is to set a subject independent threshold. However, this can change with the effectiveness of the FRS to recognize faces. Finally, it was determined that the highest threat posing FMAs were generated with the Webmorph and StyleGAN algorithms. Due to its inability to hide artefacts, Webmorph FMAs can be easily detected with TDBMADs. Unlike webmorph, StyleGAN FMAs have little to no artefacts after the morphing process. This way, they cannot be detected with traditional texture based MADs. Their detection is only possible through the use of more sophisticated methods such as employing a well-trained CNN.

There are many different face morph attack techniques and more are being created as time passes. In addition to this, not all face morphs are created entirely with algorithms. The created morphs are usually re-edited with photo editing programs such as GIMP and Photoshop, which allows the criminals to further hide their morph. Furthermore, the effects of post-processing such as scanning, printing or modification makes it even more difficult to detect FMAs. A fully functional MAD still remains to be discovered, but the results of this paper and the previous researches can be used to design a MAD system that can detect FMAs with very high accuracy. In today's world, this is crucial and necessary to assure international security as well as personal privacy.

5. Future Work

There are many possible future work that can be conducted in this field. First, there are certain simple open-source differential morphed image detection algorithms that can be testbenched in experimental designs. Although these algorithms are much less effective than the ones used in modern cybersecurity, they can provide some insight into the working mechanism of MADs. A dataset similar to the one used in this experiment can be used to design an experiment where a morphed image and a bona fide image will be provided as input to the MAD algorithm, so that the results can be evaluated. Second, different face recognition algorithms can be tested with similar methodology, so that the ways of setting the verification thresholds can be evaluated. Some face recognition algorithms might be more effective when they have a subject dependent verification threshold. This way, a general idea on the threshold determination can be concluded. Another important idea in this field is the effects of post-processing. It was briefly mentioned in this research that various post-processing effects can induce unwanted results on the face recognition system. If such effects are researched in an experiment, the ways to overcome the effects of post-processing can be devised. In real life, the morphed image is presented both in a printed and a digital medium. The differences between these two situations can be analyzed by conducting an experiment with printed and digital morphs. Another important future work that can be done in this field is to create an experimental design where the situation will be as close as possible to the real life situation. Criminals of face morph attacks never present a morphed image with obvious artefacts or low quality morphs that can easily give themselves away. Therefore, the generation of morphs between similar looking subjects and test benching the results is crucial to understand the situation. Maybe twins with very little facial differences can be used to generate morphs. In addition, the human factor should not be disregarded. A survey can be made to see whether people think an image is morphed or not. Because in border security or passport applications, security personnel are responsible for checking for FMAs. Finally, a research on an ideal morph attack detection can be made to find a MAD that can detect morphs with perfect accuracy. Many of the related work in this field use no-reference or differential MADs only by themselves. However, the usage of a no-reference MAD together with a differential one can be very useful in detecting a morph. Using the Bayes decision theory, prior probabilities can be extracted from the no-reference MAD and post probabilities can be acquired with the differential MAD. Using this post and prior probabilities, the differential MAD can make a decision. Dual usage of MAD algorithms might yield higher accuracy compared to the singular usage of MADs. Again, an experiment is necessary to understand whether this would be effective or not. The mysteries of this interesting domain remain, but with the proper research, they can gradually be revealed one by one.

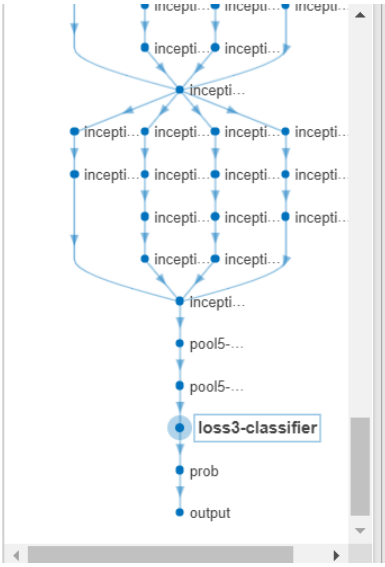
References

1. Alyssaq. (2019, June 30). Face Morpher. Retrieved from https://github.com/alyssaq/face_morpher
2. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Bayesian Decision Theory. In Pattern Classification. essay, Wiley-Interscience.
3. Flamingo, D. FaceFusion. Retrieved November 6, 2021, from http://www.wearemoment.com/FaceFusion/.
4. Mallick, S. (2021, May 5). *Face morph using OpenCV - C++ / Python: learnopencv #*. LearnOpenCV. Retrieved November 6, 2021, from https://learnopencv.com/face-morph-using-opencv-cpp-python/.
5. M. Bichsel. Automatic interpolation and recognition of face images by morphing. In Proc. of the Second Intl. Conf. on Automatic Face and Gesture Recognition. IEEE Comput. Soc. Press, 1996.
6. 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 International Workshop on Biometrics and Forensics (IWBF 2017), 2017.
7. Liu, Y.-W. A study on face morphing algorithms. Retrieved November 6, 2021, from https://ccrma.stanford.edu/~jacobliu/368Report/index.html.
8. Office, U. S. Government Accountability. "Facial Recognition Technology: Federal Law Enforcement Agencies Should Have Better Awareness of Systems Used By Employees". *www.gao.gov*. Retrieved September 5, 2021.
9. Scherhag, U., Kunze, J., Rathgeb, C., Busch, C. (2020). Face morph detection for unknown morphing algorithms and image sources: A multi‐scale block local binary pattern fusion approach. IET Biometrics, 9(6), 278–289. https://doi.org/10.1049/iet-bmt.2019.020
10. Snapp, R. Chapter 1 Pattern Classification. Retrieved November 4, 2021, from https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm.
11. Venkatesh, S., Ramachandra, R., Raja, K., Busch, C. (2020, November 3). Face Morphing Attack Generation & Detection: A Comprehensive Survey.
12. X. Yu, G. Yang and J. Saniie, "Face Morphing Detection using Generative Adversarial Networks," *2019 IEEE International Conference on Electro Information Technology (EIT)*, 2019, pp. 288-291, doi: 10.1109/EIT.2019.8834162.
13. Zienert, Naser & Grebe, Jonas & Damer, Steffen & Kirchbuchner, Florian & Kuijper, Arjan. (2019). On the Generalization of Detecting Face Morphing Attacks as Anomalies: Novelty vs. Outlier Detection. 10.1109/BTAS46853.2019.9185995.
14. Makrushin, A., & Hildebrandt, M. (2018). AMSL Face Morph Image Data Set. Retrieved December 26, 2021, from https://omen.cs.uni-magdeburg.de/disclaimer/index.php.
15. Neubert, T., Makrushin, A., Hildebrandt, M., Kraetzer, C., & Dittmann, J. (2018). Extended stirtrace benchmarking of biometric and forensic qualities of morphed face images. *IET Biometrics*, *7*(4), 325–332.https://doi.org/10.1049/iet-bmt.2017.0147.
16. DeBruine, L., & Jones, B. (2021, April 1). *Face research lab London Set*.

figshare. Retrieved December 26, 2021, from https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666

17. Faruqui, N. (2021, June 17). *Face recognition using GoogleNet*. Scholarshipin.com. Retrieved December 28, 2021, from https://www.scholarshipin.com/face-recognition-using-googlenet/
18. U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R. N. J.Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel,R. Breithaupt, R. Ramachandra, and C. Busch. Biometric systems undermorphing attacks: Assessment of morphing techniques and vulnerabilityreporting. In Intl. Conf. of the Biometrics Special Interest Group BIOSIG2017, pages 1–7, 2017.

Appendix 1: Last 9 nodes of the network are shown in the below architecture. 142nd and 144th layers of the network were replaced.



ANALYSIS RESULT

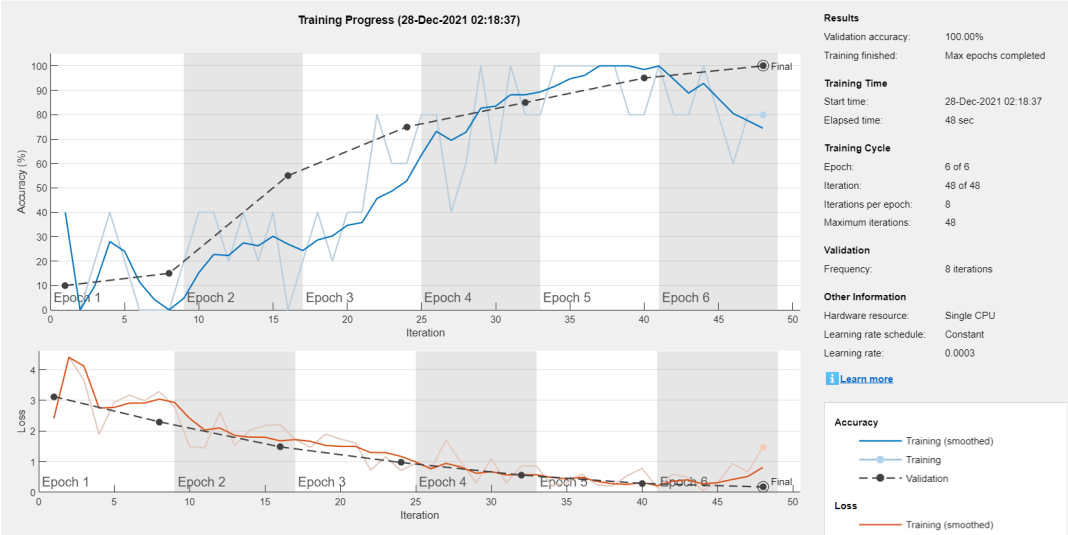
	Name	Type	Activations	Learnables
135	inception_5b-relu_5x5 ReLU	ReLU	7×7×128	-
136	inception_5b-pool 3×3 max pooling with stride [1 1]...	Max Pooling	7×7×832	-
137	inception_5b-pool_proj 128 1×1×832 convolutions with ...	Convolution	7×7×128	Weights 1×1×832×128 Bias 1×1×128
138	inception_5b-relu_pool_proj ReLU	ReLU	7×7×128	-
139	inception_5b-output Depth concatenation of 4 inputs	Depth concatenation	7×7×1024	-
140	pool5-7x7_s1 2-D global average pooling	2-D Global Average...	1×1×1024	-
141	pool5-drop_7x7_s1 40% dropout	Dropout	1×1×1024	-
142	loss3-classifier 1000 fully connected layer	Fully Connected	1×1×1000	Weights 1000×1024 Bias 1000×1
143	prob softmax	Softmax	1×1×1000	-
144	output crossentropyex with 'tench' and ...	Classification Output	1×1×1000	-

Appendix 2: Training configurations

```
Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Augmented_Training_Image.Files)/Size_of_Minibatch);
Training_Options = trainingOptions('sgdm',...
    'MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6,...
    'InitialLearnRate', 3e-4,...
    'Shuffle', 'every-epoch', ...
    'ValidationData', Augmented_Validation_Image, ...
    'ValidationFrequency', Validation_Frequency, ...
    'Verbose', false, ...
    'Plots', 'training-progress');

net = trainNetwork(Augmented_Training_Image, Layer_Graph, Training_Options);
```

Appendix 3: Training process



Appendix 4: Verification scores of the bona fide images of each subject for each trial

Subject	Verification Score				
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
1	1	0.99	1	1	0.99
2	0.98	0.91	0.97	1	0.96
3	0.73	0.77	0.76	0.81	0.8
4	0.57	0.65	0.51	0.9	0.66
5	0.65	0.56	0.59	0.56	0.42
6	0.81	0.62	0.78	0.73	0.95
7	0.66	0.67	0.58	0.9	0.70
8	0.80	0.85	0.89	0.46	0.98
9	1	0.99	0.99	1	0.99
10	0.71	0.99	0.93	0.92	0.89