A Framework to Improve the Comparability and Reproducibility of Morphing Attack Detectors

Nicolò Di Domenico, Guido Borghi, Annalisa Franco, Matteo Ferrara, Davide Maltoni
Dipartimento di Informatica — Scienza e Ingegneria (DISI)
University of Bologna
Cesena, Italy
{name.surname}@unibo.it

***Abstract*—Morphing Attack, *i.e.* the deception of Face Recognition Systems (FRS) through a face morphing process between the identity of two subjects with criminal intent, has recently emerged as a serious security threat. Due to its significance, recently several Morphing Attack Detection (MAD) systems, *i.e.* methods based on Artificial Intelligence able to automatically detect the presence of morphing, have been proposed in the literature. Unfortunately, developing, comparing, and reproducing these MAD algorithms is challenging, particularly for deep learning-based solutions, since they are usually evaluated on private datasets and the source code is not publicly released. Therefore, we observe the need for an open-source framework that aims to simplify the development of new MAD systems, in combination with their evaluation. Thus, in this paper, after a discussion about the current limits of existing studies on the MAD task, we examine the desired properties and features of this framework, with a particular focus on its modularity, usability, and effectiveness.**

***Index Terms*—Morphing Attack, Morphing Attack Detection (MAD), Single-image MAD (S-MAD), Differential MAD (D-MAD), Automated Border Control (ABC), Face Recognition Systems (FRS)**

I. Introduction

It has been shown that a subject without a criminal history, usually referred to as *accomplice*, may apply for an official document using a morphed mugshot photo that conceals the identity of a *criminal*. Indeed, through a successful *Morphing Attack* [1], [2], as represented in Figure 1, it is possible to destroy the unique link between an official document and its legitimate owner by allowing two different people to share it. In particular, several literature studies [3], [4] have demonstrated that morphed images can effectively deceive both the human expert, *e.g.* a police officer doing a visual inspection, and the currently available commercial-off-the-shelf (COTS) Face Recognition Systems (FRSs).

Therefore, the morphing attack represents a real security threat for face verification-based applications, *i.e.* systems that compare two faces in order to define if they belong to the same identity. For instance, these systems are present at Automated Border Control (ABC) gates at international airports and automatically verify the facial photo stored in the electronic Machine Readable Travel Document (eMRTD) against a live image of the person taken directly at the gate.

(a) Subject 1 (b) Morphed (c) Subject 2

Fig. 1. An example of a morphed face (central), created starting from two subjects, *i.e.* the accomplice (Subject 1) – a person without criminal records – and the criminal (Subject 2). Several studies have revealed that the morphed identity can effectively deceive both human examiners and automatic face verification-based systems, *e.g.* systems placed in Automated Border Control (ABC) gates at international airports.

Hence, the availability of robust and effective *Morphing Attack Detection* (MAD) algorithms [5] is essential to automatically detect the presence of a morphed face, and robust solutions are strongly needed by private and public institutions. Unfortunately, although several MAD approaches have been proposed in the literature in the last few years, the overall accuracy level reached so far is still unsatisfactory for effective and real-world use cases in which, for ABC systems operating in verification mode, algorithms have to ensure a False Acceptance Rate (FAR) equal to 0.1% and a False Rejection Rate (FRR) lower than 5% [6]. Indeed, despite efforts made by the scientific community, several issues hamper the effectiveness and suitability of MAD methods, as follows.

Firstly, research institutions and laboratories usually work with datasets that are not publicly released, making it challenging to evaluate the impact of training data on overall performance [7] of newly proposed MAD algorithms. Indeed, the relative novelty of the MAD task, introduced for the first time in [1], in combination with privacy issues in sharing personal data, has resulted in a lack of publicly available datasets of morphed images for training and validation that also limit the generalization capabilities of new methods.

This project received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 883356. This text reflects only the author's views, and the commission is not liable for any use that may be made of the information contained therein.

Secondly, the level of reproducibility of literature MAD methods is partially constrained by the lack of publicly available source code, usually not released by the researchers and the paper authors. In particular, despite the development of public evaluation platforms for MAD approaches, such as NIST FRVT MORPH [8] or FVC-onGoing [9], [10], a shared and standardized approach for training and testing MAD algorithms across different research laboratories has not yet been established in the literature. We observe that these aforementioned benchmarks provide an objective performance assessment by testing the submitted algorithm on sequestered datasets, *i.e.* data never seen during training and not owned by laboratories and algorithm developers, representing a valuable resource for MAD testing. However, reproducing and comparing published methods still remains a challenging task, particularly for deep learning-based solutions.

Therefore, in this paper, we discuss the proposal of an open-source framework to develop and train MAD methods, in combination with the use of public evaluation benchmarks based on sequestered data. In particular, we investigate the development of a modular framework specifically designed to support the development, training, and validation of all types of MAD systems. Its primary goal is to streamline the development and comparison of MAD systems by simplifying the usage and integration of new components, defining standard protocols, and relying exclusively on publicly available datasets, for both training and validation procedures.

II. Face Morphing

In the field of computer graphics and animation, *image morphing* is an effect that is capable of transforming one image into another through a seamless transition. This technique was originally described in [11] and used for creative tasks, such as the creation of visual effects in movies (*e.g. Willow*, 1988). However, this technique can be effectively used for a variety of applications and subjects, including human faces. Indeed, starting from two subjects it is possible to apply a *face morphing* process to obtain one or many intermediate faces, as shown in Figure 1. Since face morphing has been an active area of image processing research [12] with a wide variety of applications and scenarios, a great variety of commercial and open-source morphing tools are available, such as FaceMorpher [13], FaceFusion [14] and Sqirlz Morph [15].

Morphing algorithms can be divided into two major categories: facial landmark-based and GAN-based. GAN-based morphing algorithms employ *Generative Adversarial Networks* (GANs) such as StyleGAN [16] and MIPGAN [17] to generate the morphed image. On the other hand, landmark-based face morphing algorithms can be composed of two sequential steps applied on two input images A and B:

- Warping procedure: it is the geometric transformation needed to align the set of points in images A, B to an intermediate position, obtained by the weighted average of the two original sets of landmarks. While several warping functions have been proposed in literature [18], a common approach consists in representing the two sets of points by means of topologically equivalent triangular meshes, derived via *Delaunay triangulation* [19].
- Image blending procedure: obtained as a weighted average of the pixel intensity of the two images.

Many morphing algorithms employ an α parameter, also called *morphing factor*, which weighs the presence of the two contributing subjects in the image (in particular, the landmark positions in the warping procedure). The choice of the proper morphing factor is essential in order to fool both the human examiner (for instance, during the document issue procedure) and the FRS at ABC gates [20].

III. Morphing Attack Detection

Two families of MAD approaches can be coarsely categorized, according to the number of face images used as input: *Single-image* or *Differential* (respectively, abbreviated to S-MAD and D-MAD). In both cases, the output of a MAD system is represented by a score in the $[0, 1]$ range that indicates if one or more images are genuine *(bona fide)* or not *(morphed)*.

A. Single-image MAD (S-MAD)

Single-image MAD systems receive one image as input, then the morphing process is detected using only a single image, as depicted in Figure 2. Indeed, these methods work under the assumption that the morphing process leaves specific traces in the image, in terms of texture anomalies or visual artifacts [21], such as ghost or half-shade effects that can occur due to regions not overlapping exactly (*e.g.* hair, pupils, and nostrils), or distorted edges or shifted image areas. As S-MAD systems do not have access to additional images, this task is generally considered more challenging than D-MAD [22].

Unfortunately, a sufficiently motivated criminal can manually post-process the morphed image using off-the-shelf image editing software in order to reduce the amount and severity of the produced artifacts, thus creating a very high-quality morphed image and posing a serious challenge for S-MAD systems. Moreover, while biometric passports do include a digital copy of the photo ID of the citizen, this is always compressed in order to fit in the limited chip memory, and the photo inside the chip is often a printed and scanned version of the original; these two factors, usually combined, have the effect of drastically reducing the amount of detectable artifacts [21], [23].

Fig. 2. A typical pipeline for S-MAD systems. The input is represented by the mugshot picture of the subject typically contained in the document, and the S-MAD algorithm outputs whether the given image has undergone a morphing process. Essentially, S-MAD methods consist of detectors of the artifacts produced by the morphing procedure.

B. Differential MAD (D-MAD)

Differential MAD systems, also referred to as double-image MAD, receive a pair of images as input, and the morphing process is detected by comparing the two sources as visually summarized in Figure 3. Differently from S-MAD methods, D-MAD systems operate on the assumption that one of the two photos comes from a trusted source, *e.g.* from the camera installed in an ABC gate or from a police officer who is present when taking the subject's mugshot photo. Then, D-MAD methods usually compare the identity of a couple of images to detect the presence of morphed images and are less focused on the detection of artifacts [24].

From a general point of view, D-MAD systems can be grouped into two subcategories [25]: i) algorithms that extract and compare feature vectors (embeddings) extracted from both input images, usually through deep learning architectures trained for the Face Recognition task; ii) algorithms that try to reverse the morphing process, such as the work presented in [26], referred to as *demorphing*.

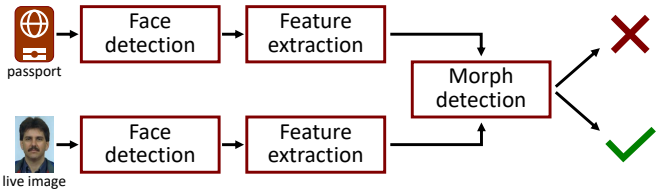

Fig. 3. A typical pipeline for D-MAD systems. By comparing the identity of the mugshot picture stored in the passport and a trusted, live-capture image of the subject, the D-MAD algorithm outputs whether the picture contained in the passport has undergone a morphing process.

C. Metrics

In order to evaluate and compare MAD systems, the most commonly-used metrics [3] are the *Bona Fide Presentation Classification Error Rate* (referred to as BPCER), which quantifies the percentage of bona fide images incorrectly classified as morphed, and the *Attack Presentation Classification Error Rate* (in short, APCER), which denotes the proportion of morphed images wrongly identified as bona fide; these two indicators are mathematically described as follows:

$$\text{BPCER}(\tau) = \frac{1}{N}\sum_{i=1}^{N} H(b_i - \tau) \tag{1}$$

$$\text{APCER}(\tau) = 1 - \left[\frac{1}{M}\sum_{i=1}^{M} H(m_i - \tau)\right] \tag{2}$$

in which τ is the score threshold on which the detection scores b_i, m_i are compared; $H(x) = \{1 \text{ if } x > 0, 0 \text{ otherwise}\}$ is defined as a step function. Moreover, we measure the BPCER with respect to a defined value of APCER, *i.e.* $\text{B}_{0.1}$, $\text{B}_{0.05}$ and $\text{B}_{0.01}$, representing the lowest BPCER with APCER $\leq 10\%$, $\leq 5\%$ and APCER $\leq 1\%$, respectively. Finally, the APCER and BPCER metrics can be plotted to create the *Detection Error Trade-off* (DET) curves to facilitate the comparison between different approaches. The *Equal Error Rate* (EER), *i.e.* the error rate for which both BPCER and APCER are equal, is usually depicted in the plot or included as a single value.

IV. Analysis on current MAD systems

As previously mentioned, different issues affect currently available MAD systems. For the sake of analysis, we group and highlight these aspects in Table I on a representative selection of MAD methods. In particular, we report if the proposed systems are tested through a dataset-wise and morphing algorithm-wise cross-validation, *i.e.* if the systems are trained and tested using different data and morphed images obtained with different approaches. Then, we indicate if each method is benchmarked on the available public platforms. Finally, in the right part of the Table, we report an analysis of the reproducibility, intended as the possibility of implementing and reproducing the method obtaining similar results by an external laboratory or institution.

Specifically, we label the level of reproducibility with three marks (low - medium - high), according to the presence of the following elements in the original paper: i) *Data*: the use of public facial datasets, such as FERET [36] or FRGC [37], to get facial images of look-alike identities used in the morphing procedure; ii) *Morphing algorithm*: the use of public morphing algorithms, such as OpenCV [38] or FaceMorpher [13] to create morphed images; iii) *Couples*: the release of the list of subjects selected from available data and used to create morphed images. In particular, it is important to know the two images that have been used in order to obtain the very same morphed face, given the original data and morphing algorithm; iv) *Code*: the release of the source code of the proposed system. It is worth noting that only the joint presence of all these elements makes a proper implementation of the method possible.

Considering Table I, we observe that none of the reported methods have been publicly released, *i.e.* the original source code is not available to other researchers. Moreover, despite the presence in several works of details about the morphing procedure, the list of exploited couples is not reported, hampering the possibility to train and evaluate the proposed method on the same data. Moreover, in some cases, the morphing algorithm used is not reported, or the morphed images have undergone manual retouching processes that cannot be replicated.

These elements, in combination with the fact that usually MAD methods are trained and tested on private datasets (*i.e.* morphed data created with private source data and/or private morphing algorithm and/or without specifying the couples exploited), hinder the comprehension, evaluation, and comparison of the newly introduced methods. From a practical point of view, it is hard to answer the following question: "*Is the performance of the proposed method improved by the data*

TABLE I

ANALYSIS OF AVAILABLE MAD SYSTEMS. FOR EACH METHOD, WE REPORT WHETHER THE METHOD IS TESTED USING DATASET-WISE AND MORPHING ALGORITHM-WISE CROSS-VALIDATIONS, IF IT IS BENCHMARKED ON PUBLIC PLATFORMS (FVC-ONGOING [10] OR NIST FRVT MORPH [8]) AND THE LEVEL OF REPRODUCIBILITY. FURTHER DETAILS ARE REPORTED IN SECTION IV.

Method	Year	Type	Features	Cross Validation		Benchmarks		Reproducibility				
				Dataset	Morph. Alg.	FVC	NIST	Grade	Data	Morph. Alg.	Couples	Code
[27]	2017	S-MAD	LBP					Low	✓	✗	✗	✗
[28]	2018	S-MAD	LBP	✓		✓		Low	✓	✗	✗	✗
[29]	2018	S-MAD	Fourier					Low	✓	✗	✗	✗
[30]	2019	S-MAD	PRNU	✓	✓	✓	✓	Medium	✓	✓	✗	✗
[31]	2021	S-MAD	Wavelets				✓	Medium	✓	✓	✗	✗
[32]	2022	S/D-MAD	Deep	✓	✓		✓	Medium	✓	✓	✗	✗
[26]	2017	D-MAD	-	✓†	✓†	✓	✓	Medium	✓	✓⋆	✗	✗
[33]	2018	D-MAD	Landmarks					High	✓	✓	✓‡	✗
[34]	2020	D-MAD	Deep	✓	✓	✓	✓	Medium	✓	✓	✗	✗
[22]	2021	D-MAD	Deep	✓	✓	✓		Medium	✓	✓⋆	✗	✗
[35]	2021	D-MAD	Mixture	✓	✓			Medium	✓	✓	✗	✗

⋆ No reproducible manual retouch on morphed images
† Not a learning method, no training data needed
‡ URL not reported in the original paper

exploited, by the method itself, or by a combination of the two?"

Furthermore, we note that the usage of private data limits the generalization capabilities of MAD systems, which usually present great accuracy only on data similar to the training set. This observation is particularly true for the more challenging S-MAD task, in which the performance significantly drops with new unseen data [10]. Finally, the use of public benchmarks, such as FVC-onGoing [9] and NIST FRVT MORPH [8], is useful in order to understand the real performance of a MAD system on sequestered datasets, *i.e.* data never seen during the training and, in general, by the researchers. Unfortunately, this information is not enough to answer the previous question.

V. MAD FRAMEWORK

In light of the points previously highlighted and discussed, we believe that a simple and unified framework is crucial for the development of new Single-image Morphing Attack Detection (S-MAD) and Differential Morphing Attack Detection (D-MAD) systems. In particular, the MAD framework should address the following elements, hiding much of the complexity that characterizes Machine and Deep Learning approaches:

- **Modularity**: the framework should be composed of different modules (such as data loading, face detection, preprocessing procedures, as detailed in Section V-A), separating the complexity and offering a single development environment for S-MAD and D-MAD approaches. Moreover, should the existing modules prove insufficient, minimal effort should be needed to implement custom functionality, whether it is a novel data augmentation stage, a different feature extractor, or a completely new model.
- **Flexibility**: all modules within the framework should rely on a single configuration file, which the user would leverage to manage and operate the entire framework. It should be trivial for the (also non-technical) end user to switch between various face detectors, modify the data augmentation pipeline, or use a different feature extractor, only changing a few lines in the configuration file.
- **Simple usage**: the framework should be designed to be deterministic, rendering the training and testing of a given model to be simple, fully reproducible, and comparable, especially if using public datasets.

A. Framework Modules

As discussed above, the framework would consist of many modules. In this Section, we identify the key modules for the implementation of new S-MAD and D-MAD methods.

The first module regards the data loading procedure, relying on the user-defined specifications in the experiment configuration file. Specifically, the user should be able to indicate one or more datasets to be employed for training and testing, providing a considerable level of flexibility, and the split ratio for training, validation, and test sets. The sum of these ratios may be smaller than 1 if the user does not wish to load the entire dataset. Once all datasets are loaded, the framework would merge the three subsets, obtaining a global training, validation, and test set.

The second module, after the loading of data, consists of the face detection operation, *i.e.* the task of identifying one or more face regions in the input image. This procedure is usually performed through a face detector, whose output is essentially a bounding box indicating the face's position inside the image represented by its top-left and bottom-right corners. Furthermore, if the face detector allows it (*e.g.* the DLib [39] face detector), facial landmarks can be extracted and embedded in the object that represents the dataset element's image. The framework should implement several widely-used face detectors, such as DLib [39] (particularly used in biometrics), OpenCV [40] and MTCNN [41].

The third module, only applicable to the training set, should contain the data augmentation procedures. This module must be optional since it is not certain that MAD systems need

data augmentation techniques, and may be skipped during model training. The augmentation pipeline comprises multiple sequential steps useful in the MAD task, such as image resizing, horizontal flip, grayscale filter, compression (*e.g.* JPEG), jittering, and other similar operations useful to prevent overfitting phenomena and improve generalization capabilities. Moreover, it is important also to include a simulation of the printing and scanning process (P&S), which is of particular significance in MAD methods applied on printed images [42], [43]. Moreover, in the case of a D-MAD algorithm, the user may want to apply a specific augmentation step to only one of the images in every dataset element.

Another optional module is responsible for feature extraction, *i.e.* the task of using a feature extractor to extract significant attributes from input images. A feature extractor can be defined as a pre-trained network capable of extracting features related to the training task: for instance, in the case of models trained for Face Recognition, features related to the subject's identity can be provided. Alongside this, a feature extractor can also be a mathematical operation applied to the input images: this is the case when a Fourier transformation is utilized to extract magnitude spectrums. Other features proposed in the literature (especially in the MAD task) and that can be implemented in the framework are reported in Table I, such as Photo Response Non-Uniformity (PRNU) [30], [44], wavelets [31], [45] and Fourier transform [46].

One of the most important modules of the framework is the one responsible for the proper training of the MAD model. Thus, the aforementioned configuration file should be divided into two sections: model definition and training. In the model definition section, the user is required to define the model for the experiment. In the training section, the user must specify all the information needed to train the model. The training section's contents vary depending on the selected model since different models require different training configuration arguments: the user can set the key elements for the training of Deep Learning models, such as the number of epochs, the batch size, the loss function, and the optimizer. We observe that the most common loss function and optimizers used for the MAD task are Binary Cross-Entropy (BCE) loss (the two classes are referred to as morphed and bona fide), Adam [47], and the Stochastic Gradient Descent (SGD). Of course, it should be possible to define also custom loss functions and optimizers. Other features such as checkpoints (model weights saving after each epoch), early stopping (to contrast overfitting phenomena), and experiment logging must be implemented to have an effective training loop. In particular, training events can be captured via callbacks in the following (but not only) steps: before/after training, before/after training/validation epoch, and before/after training/validation step.

Finally, a module responsible for the evaluation is mandatory: several built-in metrics commonly used in literature [30] while developing MAD systems should be implemented, including the classification accuracy, Equal Error Rate (EER), Bona fide Presentation Classification Error Rate (BPCER) at one or many user-defined Attack Presentation Classification Error Rates (APCERs), as detailed in Section III-C. In order to simplify cross-validations with datasets and morphing algorithms, the framework should allow defining logical test sets, and metrics are consequently reported for each distinct testing group, in addition to the whole test set. Therefore, users can have separate metric values divided by the dataset, algorithm, morphing factor, or possible combinations of these. Finally, the framework should save the metrics and computed scores for each testing group to text files, allowing for easy human inspection. Additionally, the metrics for each testing group can be dumped into a formatted file (*e.g.* JSON), making them more readily accessible by automated scripts that can parse such file format.

VI. Conclusion

In this paper, we have highlighted and analyzed issues that commonly affect literature Morphing Attach Detection systems, limiting their effectiveness, generalization capabilities and reproducibility. Therefore, we have discussed the development of a framework aimed to simplify and improve the comparability and reproducibility of newly proposed MAD systems in the literature. We believe that this work can be useful to discuss the challenges that should be addressed in future works related to the MAD research field.

References

[1] M. Ferrara, A. Franco, and D. Maltoni, "The magic passport," in *IEEE International Joint Conference on Biometrics, Clearwater, IJCB 2014, FL, USA, September 29 - October 2, 2014*. IEEE, 2014, pp. 1–7.

[2] M. Ferrara and A. Franco, *Morph Creation and Vulnerability of Face Recognition Systems to Morphing*. Springer International Publishing, 2022, pp. 117–137.

[3] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face recognition systems under morphing attacks: A survey," *IEEE Access*, vol. 7, pp. 23 012–23 026, 2019.

[4] U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R. N. Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel *et al.*, "Biometric systems under morphing attacks: Assessment of morphing techniques and vulnerability reporting," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2017, pp. 1–7.

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

[6] M. Ferrara and A. Franco, "Morph creation and vulnerability of face recognition systems to morphing," in *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*. Springer International Publishing Cham, 2022, pp. 117–137.

[7] G. Borghi, G. Graffieti, A. Franco, and D. Maltoni, "Incremental training of face morphing detectors," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, 2022, pp. 914–921.

[8] National Institute of Standards and Technology. NIST FRVT Morph. [Online]. Available: https://pages.nist.gov/frvt/html/frvt_morph.html

[9] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue, "Fingerprint and on-line signature verification competitions at icb 2009," *Advances in Biometrics*, pp. 725––732, 2009.

[10] Biolab. FVC-onGoing. [Online]. Available: https://biolab.csr.unibo.it/fvcongoing/

[11] D. B. Smythe, "A two-pass mesh warping algorithm for object transformation and image interpolation," *Rapport technique*, vol. 1030, p. 31, 1990.

[12] G. Wolberg, "Image morphing: a survey," *The visual computer*, vol. 14, no. 8-9, pp. 360–372, 1998.

[13] A. Quek, "FaceMorpher morphing algorithm." [Online]. Available: https://github.com/alyssaq/face_morpher

[14] FaceFusion, "Facefusion," http://www.wearemoment.com/FaceFusion/, accessed: 2022-11-30.

[15] xiberpix, "Sqirlz morphing algorithm," https://sqirlz-morph.it.uptodown.com/windows, accessed: 2022-11-30.

[16] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

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

[18] G. Wolberg, *Digital Image Warping*, ser. Systems. IEEE Computer Society Press, 7 1990.

[19] L. P. Chew, "Constrained delaunay triangulations," in *Proceedings of the Third Annual Symposium on Computational Geometry*, ser. SCG '87. Association for Computing Machinery, 1987, pp. 215—-222.

[20] 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)*. IEEE, 2019, pp. 1–5.

[21] G. Borghi, A. Franco, G. Graffieti, and D. Maltoni, "Automated artifact retouching in morphed images with attention maps," *IEEE Access*, vol. 9, pp. 136 561–136 579, 2021.

[22] G. Borghi, E. Pancisi, M. Ferrara, and D. Maltoni, "A double siamese framework for differential morphing attack detection," *Sensors*, vol. 21, no. 10, p. 3466, 2021.

[23] U. Scherhag, J. Kunze, C. Rathgeb, and C. Busch, "Face morph detection for unknown morphing algorithms and image sources: a multi-scale block local binary pattern fusion approach," *IET Biometrics*, vol. 9, no. 6, pp. 278–289, 2020.

[24] N. Di Domenico, G. Borghi, A. Franco, and D. Maltoni, "Combining identity features and artifact analysis for differential morphing attack detection," in *Proceedings of the 22nd International Conference of Image Analysis and Processing, Udine, Italy*, 2023.

[25] U. Scherhag, "Face morphing and morphing attack detection," Ph.D. dissertation, Technical University of Darmstadt, Germany, 2021.

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

[27] R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch, "Face morphing versus face averaging: Vulnerability and detection," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 555–563.

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

[29] L.-B. Zhang, F. Peng, and M. Long, "Face morphing detection using fourier spectrum of sensor pattern noise," in *2018 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2018, pp. 1–6.

[30] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl, "Detection of face morphing attacks based on PRNU analysis," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 4, pp. 302–317, 2019.

[31] P. Aghdaie, B. Chaudhary, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Attention aware wavelet-based detection of morphed face images," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2021, pp. 1–8.

[32] I. Medvedev, F. Shadmand, and N. Gonçalves, "Mordeephy: Face morphing detection via fused classification," *arXiv preprint arXiv:2208.03110*, 2022.

[33] U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch, "Detecting morphed face images using facial landmarks," in *Image and Signal Processing: 8th International Conference, ICISP 2018, Cherbourg, France, July 2-4, 2018, Proceedings 8*. Springer, 2018, pp. 444–452.

[34] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch, "Deep face representations for differential morphing attack detection," *IEEE transactions on information forensics and security*, vol. 15, pp. 3625–3639, 2020.

[35] S. Lorenz, U. Scherhag, C. Rathgeb, and C. Busch, "Morphing attack detection: A fusion approach," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–7.

[36] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, 1998.

[37] P. J. Phillips *et al.*, "Overview of the face recognition grand challenge," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005.

[38] S. Mallick, "Face morph using OpenCV — C++ / Python." [Online]. Available: https://learnopencv.com/face-morph-using-opencv-cpp-python/

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

[40] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[41] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[42] K. Raja, S. Venkatesh, C. Busch *et al.*, "Transferable deep-cnn features for detecting digital and print-scanned morphed face images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 10–18.

[43] M. Ferrara, A. Franco, and D. Maltoni, "Face morphing detection in the presence of printing/scanning and heterogeneous image sources," *IET Biometrics*, vol. 10, no. 3, pp. 290–303, 2021.

[44] L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch, "Prnu-based detection of morphed face images," in *2018 International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2018, pp. 1–7.

[45] P. Aghdaie, B. Chaudhary, S. Soleymani, J. Dawson, and N. Nasrabadi, "Detection of morphed face images using discriminative wavelet subbands," in *2021 IEEE International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2021, pp. 1–6.

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

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980