Alma Mater Studiorum - Università di Bologna
CAMPUS DI CESENA

Dipartimento di Informatica - Scienza e Ingegneria
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Revelio: a Modular and Effective Framework for Reproducible Training and Evaluation of Morphing Attack Detectors

Tesi di laurea in
Visione Artificiale e Riconoscimento

Relatore:
Guido Borghi

Presentata da:
Nicolò Di Domenico

Correlatori:
Annalisa Franco
Matteo Ferrara

Anno Accademico 2021/2022

To my family and old friends:
without your continued support,
I would have never done it.

To my new friends:
we started this journey together
and we are finishing it still together.

To the researchers at West Virginia University:
you are my main source of motivation
for completing this thesis.

Abstract

Morphing Attack, *i.e.* the possibility of eluding face verification systems through a facial morphing operation between a criminal and an accomplice, has recently emerged as a serious security threat. Despite the importance of this kind of attack, the development and comparison of Morphing Attack Detection (MAD) methods is still an arduous task, mainly due to the scarcity of publicly-available datasets and the failure of the internal ones to accurately reflect the problem's complexity; these two causes combined lead to low generalization capabilities and challenges in comparing the different MAD approaches proposed in the literature. Therefore, in this thesis, we propose and publicly release *Revelio*, a flexible and modular framework for the reproducible development and evaluation of both single-image (S-MAD) and differential (D-MAD) systems. Then, we conduct a review of the datasets exploited in the literature and introduce two new ones, namely *ChiMo* and *FEI*. Moreover, we introduce a new metric useful for summarizing and simplifying the comparison of diverse approaches across different datasets, named Weighted Average Error across Datasets (WAED), and conduct a review of the publicly available benchmarks used to test algorithms for this task. Besides, an extensive analysis of several state-of-the-art approaches through Revelio is performed, comparing several literature methods and thus deeply analyzing the main challenges in the MAD task. Finally, by exploiting Revelio features, a new model is proposed to improve the state of the art on SOTAMD single-image and double-image benchmarks.

Contents

Chapter 1

Introduction

In this Chapter, we present the necessity of creating *Morphing Attack Detection* (MAD) algorithms to defend against the emerging threat represented by *Morphing Attacks* on passport photos. Then, we discuss the two families of MAD approaches: *Single-image* (S-MAD) or *Differential* (D-MAD). Finally, we formally introduce the face morphing algorithm.

1.1 Morphing attack

Figure 1.1: An example of a morphed face (central), created starting from two subjects (on both sides).

Through a *Morphing Attack* [28, 23] an official document can be shared across two different people, destroying the unique link between the document and its real owner.

In practice, a subject with no criminal records *(accomplice)* might apply for an official document using a morphed mugshot photo that hides the identity of a *criminal*, as shown in Figure 1.1. Indeed, several studies [70, 65] have shown that morphed images can be effectively used to fool both the human control, *e.g.* a police officer, and the currently available *commercial-off-the-shelf* (COTS) *Face Recognition Systems* (FRSs).

For these reasons, the morphing attack represents a serious and concrete security threat for identity verification-based applications, such as the *Automated Board Control* (ABC) gates at international airports where the facial photo stored in the *electronic Machine Readable Travel Document* (eMRTD) is automatically verified against the live acquired image of the document owner. Moreover, it has been proven that this attack was used at least once for circumventing face recognition systems [78, 46]. Therefore, the availability of *Morphing Attach Detection* (MAD) methods [59], *i.e.* systems that are able to automatically detect the presence of a morphed face in images, is strongly needed by public and private institutions and has raised the interest of researchers belonging to different areas [76]. The difficulty in solving this task is also exacerbated by the fact that different morphing algorithms may produce very different results in terms of quality and presence of artifacts, as shown in Figure 3.1.

Despite the relevant number of approaches proposed in the literature in the last years, the accuracy level reached so far by MAD systems is still unsatisfactory for deployment in real-world applications. Furthermore, a standardized way of training and testing MAD algorithms has yet to emerge in the literature. The research community devoted some efforts to the development of public evaluation platforms for MAD approaches, such as NIST FRVT MORPH [48] or FVC-OnGoing [19, 5], where the performance can be objectively assessed by supervised testing on sequestered datasets, *i.e.* data never seen during training and not owned by laboratories and algorithm developers. These benchmarks represent a valuable resource for MAD testing. However, it is worth noting that reproducing and comparing published methods is still a challenging task, especially for deep learning-based solutions. This issue probably originates from the relative novelty of the MAD task, introduced for the first time in [28], which determines two main consequences:

- the lack of publicly available datasets of morphed images on which to train and validate the proposed methods, also due to privacy issues;
- the lack of publicly available source code of the MADs proposed in the literature.

In this scenario, each research laboratory or institution usually works on its own data, thus making it difficult to evaluate the impact of the training data

on the overall MAD performance [8] and severely limiting the reproducibility of algorithms and results. We believe that the use of public evaluation benchmarks based on sequestered data, in combination with a shared framework on which publicly released datasets are exploited to develop and train newly proposed MAD methods, can help to improve the quality level of contributions and understanding in the morphing research field.

Relying on these considerations, this thesis proposes *Revelio*, a modular framework aimed at providing shared and effective support for MAD systems development, training, and validation. Revelio has been explicitly designed to reduce the efforts needed for the development and comparison of MAD systems, with particular attention to simplifying the usage and integration of new components, defining common protocols, and relying only on publicly available datasets, for both training and validation procedures. Together with this thesis, we publicly release the source code and the official documentation[1] of the framework. The released framework already includes several literature algorithms frequently used in MAD system development.

In order to test the features of Revelio, we conducted an extensive experimental validation to deeply analyze and compare the performance of several deep learning-based MAD solutions, also proving that Revelio allows the training of state-of-the-art detectors in a straightforward and simple manner. For the sake of reproducibility, all experiments are carried out on publicly released or reproducible datasets. We believe that, in this way, this work can be a useful reference for future research works, analysis, and investigations on MAD techniques.

Revelio is designed to support and speed up the development of both *Single-image Morphing Attack Detection* (S-MAD) and *Differential Morphing Attack Detection* (D-MAD) algorithms, both introduced in Section 1.2.

1.2 Types of Morphing Attack Detectors

Generally, the output of a MAD system is represented by a score that indicates if one or more images are genuine *(bona fide)* or not *(morphed)*.

Two families of approaches can be coarsely categorized, focusing on the number of face images used as input:

- *Single-image methods* (S-MAD), also referred as *no-reference* or *forensic* methods;
- *Differential methods* (D-MAD), also referred as *two-images-* or *pair*-based systems.

[1] `https://miatbiolab.csr.unibo.it/revelio-framework`

1.2.1 Single-image MAD (S-MAD)

These systems receive only one image as input and the morphing process is detected using only a single image, as depicted in Figure 1.2. Depending on the environment these systems are installed in, their input can be:

- the mugshot photo presented to the police officer during the enrollment procedure;
- the face image available during the verification procedure, *i.e.* the image ready from the eMRTD during the controls at ABC gates.

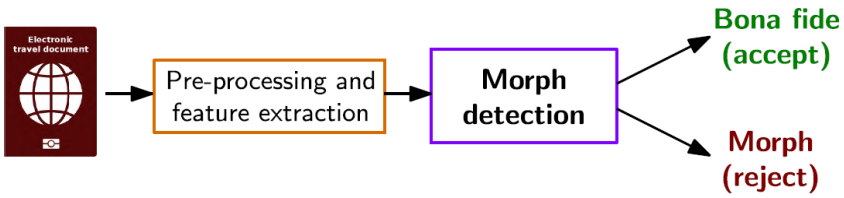


Figure 1.2: A typical pipeline for S-MAD systems. The input is represented by the mugshot picture of the subject, and the S-MAD algorithm outputs whether the given image has undergone a morphing process. Image source: [70].

In both cases, the output is usually a score or a prediction that directly reveals whether the image presents anomalies that can be traced back to a morphing process.

These methods work under the assumption that the morphing process leaves specific traces in the image, in terms of texture anomalies or visual artifacts, 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. However, this assumption can be broken by a sufficiently motivated criminal, who 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.

Finally, 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 [69]. The effects of both factors can be observed in Figure 3.7.

1.2.2 Differential MAD (D-MAD)

These systems (whose pipeline can be summarized in Figure 1.3) receive a pair of images as input, and the morphing process is detected by comparing the two sources. D-MAD systems operate on the assumption that one of the two photos must come from a trusted source, *i.e.* from the camera installed in an ABC gate or from a police officer who is present when taking the subject's mugshot photo.

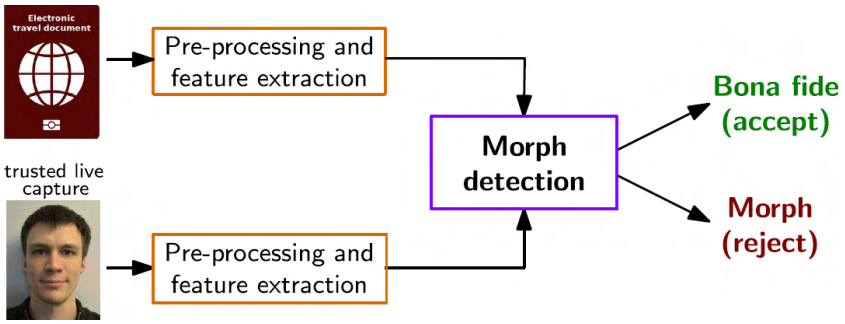


Figure 1.3: A typical pipeline for D-MAD systems. By comparing the mugshot picture stored in the passport and a trusted, live-capture image of the subject, the D-MAD algorithm outputs whether the picture contained in the passport has undergone a morphing process. Image source: [70].

These systems can be deployed in two scenarios:

- *during passport issuance*: the trusted image is the live acquisition of the face, while the provided mugshot photo represents the untrusted one;
- *during controls at ABC gates*: the trusted image is the live acquisition during the automatic face verification procedure, while the untrusted one is the image stored in the eMRTD.

D-MAD systems can be grouped into two subcategories [63]:

- the first category contains algorithms that compare *feature vectors* extracted from both input images;
- the second category relies on *demorphing*, *i.e.* algorithms that try to reverse the morphing process, as presented in [26].

1.3 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 [75] and was used for the first time by *Industrial Light and Magic*, a motion picture visual effects company based in the United States, for the movie *Willow* (1988).

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 1.1.

Morphing algorithms can be divided into two major categories: *landmark-based* and *GAN-based*. GAN-based morphing algorithms employ *Generative Adversarial Networks* (GANs) such as StyleGAN [36] to generate the morphed image. On the other hand, landmark-based face morphing algorithms can be composed of two sequential steps:

- the application of a *warping function* $w_{B \to A}$, which expresses the geometric transformation required to align the set of points in image B to the ones in A;
- *image blending*, simply 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 transformations between the two images.

Being landmark-based morphing algorithms more common, we primarily focus on those.

Formally, the face morphing process that outputs an intermediate morphed image I_α starting from two images I_0 and I_1 is defined as follows. Given the corresponding face landmarks (*e.g.* eye corners, nose tip, etc.) $P_0 = \{\mathbf{u}_i, i = 1, \ldots, N\}$ and $P_1 = \{\mathbf{v}_i, i = 1, \ldots, N\}$, respectively for I_0 and I_1, the intensity of a pixel in position $\mathbf{p}$ for the morphed image I_α can be computed using Equation 1.1:

$$I_\alpha(\mathbf{p}) = (1 - \alpha) \cdot I_0\left(w_{P_\alpha \to P_0}(\mathbf{p})\right) + \alpha \cdot I_1\left(w_{P_\alpha \to P_1}(\mathbf{p})\right) \tag{1.1}$$

In Equation 1.1, P_α is the set of landmark positions obtained by linearly interpolating the corresponding points in P_0 and P_1, as specified in Equation 1.2, whose visual explanation can be found in Figure 1.4:

$$P_\alpha = \{\mathbf{r}_i \mid \mathbf{r}_i = (1 - \alpha) \cdot \mathbf{u}_i + \alpha \cdot \mathbf{v}_i, \mathbf{u}_i \in P_0, \mathbf{v}_i \in P_1\} \tag{1.2}$$

While several warping functions $w_{B \to A}$ have been proposed in literature [82], a common approach consists in representing the two sets of points by means of topologically equivalent triangular meshes, derived via *Delaunay triangulation* [13].

While typically the α factor is the same for both face warping and image blending, in [24] an alternative version of Equation 1.1 is presented, where α is

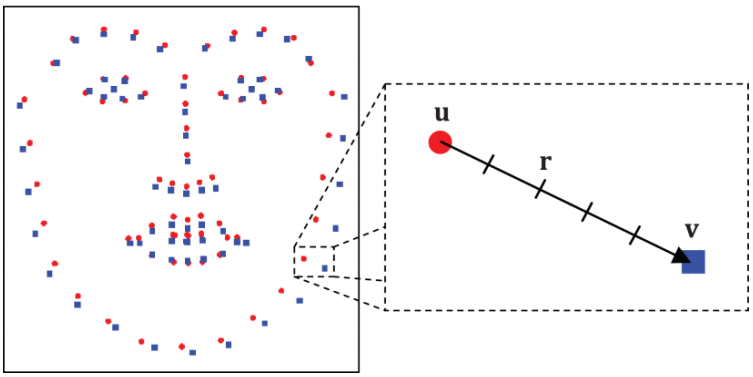


Figure 1.4: A visual explanation of Equation 1.2: the red circles are the landmarks in P_0, while the blue squares are the landmarks in P_1. On the right, it is possible to see a pair of corresponding points $\mathbf{u} \in P_0$ and $\mathbf{v} \in P_1$ with the linear interpolation $\mathbf{r} \in P_\alpha$ with $\alpha = 0.4$. Image source: [26]

replaced with two distinct values, α_W, and α_B, which represent the warping and blending factors, respectively.

$$I_\alpha(\mathbf{p}) = (1 - \alpha_B) \cdot I_0\left(w_{P_{\alpha_W} \to P_0}(\mathbf{p})\right) + \alpha_B \cdot I_1\left(w_{P_{\alpha_W} \to P_1}(\mathbf{p})\right) \tag{1.3}$$

The resulting Equation 1.3 can be used to better evaluate the importance of face warping and image blending. Indeed, given the subjects in Figure 1.5, it is possible to construct a grid consisting of the resulting face morphs by varying α_W and α_B, as shown in Table 1.1.

Figure 1.5: Images I_0 and I_1 used to generate Table 1.1. Image source: [24].

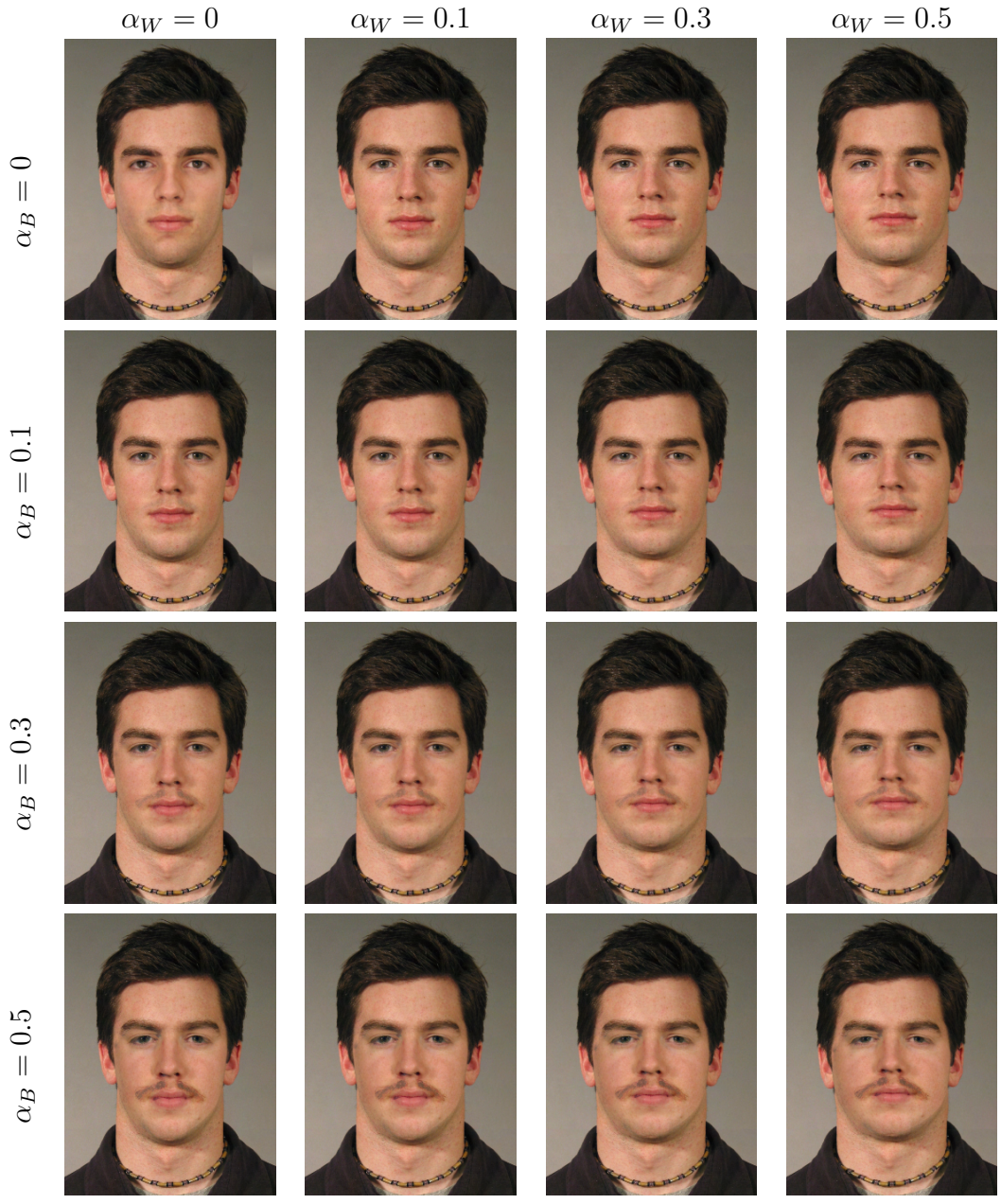


Table 1.1: Morphed images obtained with different warping and blending factors. Image source: [24].

Chapter 2

Revelio framework

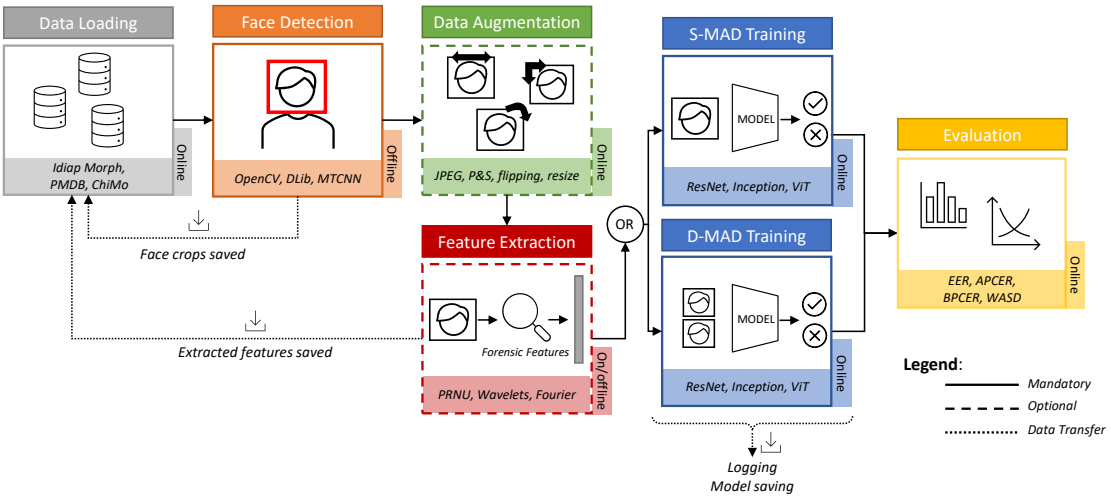


Figure 2.1: Overview of the *Revelio* framework. As depicted, the framework is modular and is mainly based on six different blocks, to simplify the development of new S-MAD and D-MAD systems and also expand the functionalities of the framework itself. Each block can be run in an online or offline manner and its presence is mandatory or optional in the final pipeline (see Chapter 2).

The design of the Revelio framework is based on the conviction that a simple and shared platform is a key element in order to develop better S-MAD and D-MAD systems. Therefore, the framework is designed to be modular and flexible, while abstracting away most of the complexity that is typical of Machine and Deep Learning approaches, such as dataset loading, definition and implementation of the data processing pipeline, model training, and finally performance evaluation according to different metrics.

The modular structure of the framework is depicted in Figure 2.1. All these

modules rely on a single YAML configuration file, through which the user manages and handles the whole framework: its main sections are included and discussed in the following paragraphs. Indeed, it is trivial for the end user to swap between different face detectors, change the data augmentation pipeline, or use a new feature extractor: only a few lines in the experiment configuration file are needed. In addition, once a seed is specified in the configuration file, the framework is designed to be deterministic: therefore, if all datasets are available, the training and testing of a given model are fully reproducible and comparable. Moreover, should the already built-in modules not be enough, little effort is needed to implement a custom functionality (be it a new data augmentation step, a new feature extractor, or a whole new model) which is then ready to be used in new experiments. This is made possible by the integrated plugin system, which allows the loading of Python files containing the new modules that can then be invoked by the experiments that require them.

In the following Sections, further details for each module are briefly reported and discussed.

2.1 Data loading

As the name suggests, this module is responsible for loading into memory all the required data, organized in datasets, according to the user-defined specifications in the experiment configuration file.

The user can specify one or many datasets to be used for training and testing, and this can be done with a great level of flexibility. Indeed, the user is required to specify only the dataset's name, root path, and split ratio for training, validation, and test sets, as detailed in Listing 2.1; while more advanced settings for loading a dataset are available, they are not strictly required and are either inferred or some sensible default values are employed. The code to be used to correctly load a dataset into memory is specified by a *dataset loader*, which takes the dataset root path as input and returns a list of dataset element descriptors: these simple objects contain only the path to one (S-MAD) or many (D-MAD) images, alongside the class the element belongs to (either bona fide or morphed).

When loading each dataset, the list of all dataset element descriptors is randomly split following the training/validation/test ratios that the user expressed in the configuration file for that dataset. By design, the sum of these ratios can be less than 1 (*i.e.* if the user does not want to load entirely a dataset). After all datasets are loaded, the framework merges together the three subsets, thus obtaining a global training, validation, and test set.

Due to the fact that this stage operates on the dataset as a whole, this process cannot be parallelized; however, the following stages can be significantly sped up

by having multiple workers processing disjoint slices of the dataset in parallel.

```yaml
datasets:
  - name: dataset_name
    path: /directory/to/dataset_root
    split:
        train: 0.7
        val: 0.1
        test: 0.2
    testing_groups:
      - testing-group-1
      - testing-group-2
    loader:
        name: MyCustomLoader
        args:
            arg1: value1
            arg2: value2
  - ...
```

Listing 2.1: Configuration of the *Data Loading* module: among different settings, it is possible to set a specific loader, defining the splits used in training, validation and testing procedures. While this is a complete configuration example, some configuration options are either optional or inferred if unspecified.

2.2 Face detection

The next module of the framework is aimed at localizing the face in each image according to a specific detection algorithm. The output of the face detector is a bounding box $((x_{TL}, y_{TL}), (x_{BR}, y_{BR}))$, which indicates the position of the face inside the image through the use of top-left (TL) and bottom-right (BR) corner coordinates. Furthermore, if the face detector supports it (*e.g.* the DLib [37] face detector), facial landmarks are extracted and embedded into the object representing the dataset element's image.

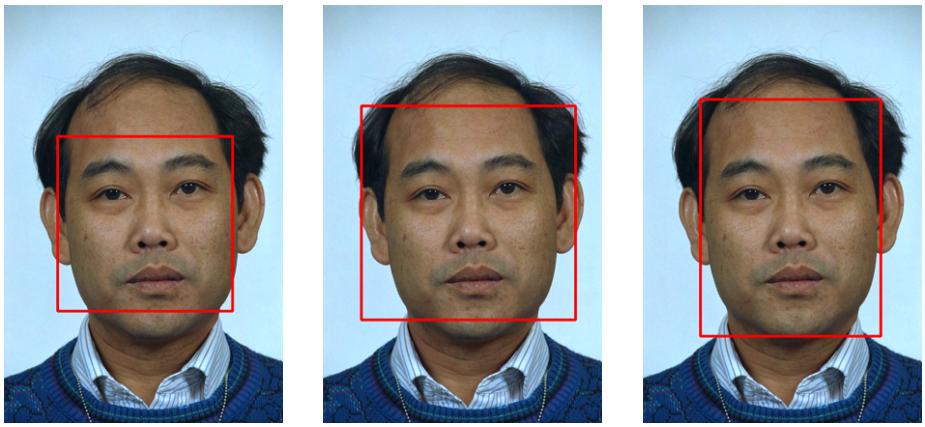
Since the face detection and landmarks extraction processes can be particularly time-consuming, this stage is carried out offline and its output is stored for each image, so that if the face detector's parameters do not change, its results will be loaded instead of being computed from scratch. We observe the choice of the face detection algorithm is a key element in the MAD task, due to the fact that each face detector produces different bounding boxes, as shown in Figure 2.2, thus potentially affecting the classification performance of the model.

From an implementation point of view, three face detectors widely used in biometrics are already available in the framework: DLib [37], OpenCV [81] and MTCNN [86].

```
face_detection:
    output_path: /directory/to/face_detection_output
    algorithm:
        name: detector_name
        args:
            arg1: value1
            arg2: value2
            ...
```

Listing 2.2: Configuration of the *Face Detection* module.

Listing 2.2 shows how the face detector can be chosen in the experiment's configuration file. The face detector's arguments are dependent on the employed algorithm: for instance, the DLib detector allows an optional argument to specify the path of the landmark detector to use, while the MTCNN detector does not have any arguments to set.



(a) DLib (b) OpenCV (c) MTCNN

Figure 2.2: Comparison between the bounding boxes generated by three different face detectors, *i.e.* DLib [37], OpenCV [81] and MTCNN [86].

2.3 Data augmentation

The following stage is applied only to the training set; validation and test sets will always skip this stage. This module is optional and can be skipped during

model training. The augmentation pipeline is composed of multiple sequential steps, each of them associated with a probability of execution on a given input element. Moreover, especially when implementing a D-MAD algorithm, the user may want to apply a certain augmentation step to only one of the images in each dataset element (*e.g.* a grayscale filter to the live-capture image and a JPEG compression on the suspected morphed image). Due to the stochastic nature of the data augmentation stage, its output is never cached. From an implementation point of view, the *Revelio* framework has already coded data augmentation procedures regarding the resize and the compression of the input data, in combination with the simulation of the printing and scanning process (P&S) that, as highlighted below, assume particular importance in the MAD task [58, 27].

Listing 2.3 shows a minimal example of a data augmentation pipeline composed of two steps: the first one applies a simulated printing and scanning process [27] to approximately half the training elements, while the second one applies a JPEG compression so that each image is under the specified number of bytes while retaining the maximum possible quality. If the probability is not specified, the step is by default applied to all the training elements.

```yaml
augmentation:
    enabled: true
    steps:
      - uses: print_scan
        probability: 0.5
      - uses: jpeg_compression
        probability: 0.5
        args:
            max_bytes: 15000
      - ...
```

Listing 2.3: Configuration of the *Data Augmentation* module. As an example, it is reported the Print & Scanned procedure, applied with a probability of 50% on input images, followed by the JPEG compression with a maximum size of 15kB.

2.4 Feature extraction

In this module, a feature extractor is used to extract features from input images. A feature extractor can be defined as a pre-trained network, able to extract features related to the training task: this is the case, for instance, of models trained for Face Recognition, which provide features related to the subject's identity. Besides, a

feature extractor can be also a mathematical procedure computed on input images: for instance, this is the case in which a Fourier transformation is applied.

Since the framework cannot know in advance how the extracted features will be used, the computed features are inserted as values in a per-image dictionary whose keys are the names of the algorithms used. These features' dictionaries are then made available to the MAD model, which is ultimately responsible for combining and using the extracted features accordingly. As feature extraction can be rather computationally expensive, this stage can be cached. However, as it is executed after the data augmentation stage (whose output is stochastic), the feature extraction results cannot be cached if any data augmentation is applied.

Inspired by methods that exploit forensic features in their implementation, PRNU [15, 68], Wavelets [1, 2] and Fourier [85] features have been implemented and tested in Revelio, as discussed in the following.

```
feature_extraction:
    enabled: true
    output_path: /directory/to/feature_extraction_output
    algorithms:
      - name: feature_extractor
        args:
            arg1: value1
            arg2: value2
            ...
      - ...
```

Listing 2.4: Configuration of the *Feature Extraction* module.

Listing 2.4 shows an example of how it is possible to configure one or multiple feature extractors to be applied to every image of each dataset element.

Analogously to the face detection section in the configuration file, each feature extractor has its own set of arguments that can be set. For instance, some feature extractors require a fixed-size image in order to produce a fixed-size feature vector/matrix, so they require both a target width and height to resize all images. Other feature extractors may not have this constraint, so those arguments would not be available.

2.5 Data preprocessing

Finally, just before feeding the images to the model, the last stage of the pipeline is *data preprocessing*, which is responsible for manipulating all the images across

training, validation, and test sets. As many models require images with a well-defined shape, this stage is particularly useful for resizing them so that they have the same shape in order to be fed into the model; another common use case for this stage is normalizing the images using specific mean and variance vectors.

The data preprocessing stage has many resemblances with the data augmentation stage, but there are some key differences: firstly, each preprocessing step is by default executed on all datasets instead of just the training set with a user-defined probability; lastly, it is applied to all images of a given dataset element.

If necessary, it is possible to have certain preprocessing steps be executed only on certain datasets: this feature can be helpful when an augmentation step is always applied with random parameters to the training set, and the same step must also be applied in a more controlled way on both the validation and test sets.

2.6 Model training

The next stage of the pipeline is responsible for the training of the MAD model. The configuration file is split into two main sections: model definition and training. In the former, the user must specify which model to adopt for the experiments; in the latter, the user must provide all the information required in order to train the model. The training section contents vary according to the model used, as different models have different training configuration arguments.

While the framework theoretically supports any type of model that can be trained and output predictions as *PyTorch* tensors, particular attention has been dedicated to deep learning-based binary classification models to discriminate bona fide and morphed images. Once the model is loaded into the specified device (either CPU or GPU), the training process starts. The user can specify the number of epochs and the batch size, and the framework will automatically split the datasets into batches. The loss function and the optimizer are also specified by the user in the configuration file, and the framework will automatically create the corresponding objects. Revelio has already implemented the most common loss functions and optimizers, such as Binary Cross-Entropy (BCE) loss, Adam, and SGD, but it is also possible to specify custom ones by simply implementing them as plugins.

In order to have a leaner training loop, some extra features such as checkpoints, early stopping, and experiment logging tools have been implemented as *callbacks*, which are objects that react to certain events inside the training loop. The framework has some callbacks already implemented, such as the one to save the model's weights at the end of each epoch, and the callback which stops the training process if the validation loss does not improve for a certain number of epochs. For the logging, a specific callback logs the various metrics through *Tensorboard*[1]. It

[1] `www.tensorflow.org/tensorboard`

is also possible to implement custom callbacks, to extend the functionalities of the framework. In total, there are 10 events that can be captured via callbacks: before/after training, before/after training/validation epoch, and before/after training/validation step.

Finally, training metrics are of prime importance when training a model. Generally, metrics are stateful objects which, after being initialized, are updated after every step by providing two tensors, respectively containing the expected and the predicted scores; as soon as the metric's state is updated, its value can be computed. Revelio comes with several built-in metrics which are widely used in literature [68] when developing MAD systems, also described in Section 3.5. Indeed, classification accuracy, *True Positive Rate* (TPR), *True Negative Rate* (TNR), *Equal Error Rate* (EER), and *Bona fide Presentation Classification Error Rate* (BPCER) at one or many user-specified *Attack Presentation Classification Error Rate* (APCER) (BPCER@APCER) are already present and ready to use, as detailed in the following.

Listing 2.5 shows how the experiment can be configured in Revelio. In the `model` section, the user specifies which model to use and sets any of its custom arguments, that vary according to the chosen model. The `training` section's contents are dependent on the type of model that is used. For instance, if the model is a neural network, there are several arguments to be set. Firstly, the user must specify the number of epochs to train the model with; moreover, an optimizer must be specified and at least its learning rate must be provided; finally, the user must select which loss function should be used. Finally, the user can specify an arbitrary number of callbacks, by specifying their name and potentially their arguments.

2.7 Evaluation

Revelio provides a way of defining logical test sets called *testing groups*, and metrics are then reported for each unique testing group, in addition to the whole test set. This way, the user can have separate metrics' values divided by dataset, algorithm, morph level, or any possible combination of these. However, some metrics (*e.g.* EER) cannot be computed if all images of a given group belong to the same class (either bona fide or morphed), so it is essential that each testing group contains at least some bona fide and morphed images.

Finally, as shown in Listing 2.6, the framework saves the metrics and computed scores (separated by their true label) for each testing group to text files, so that they can be easily inspected by humans. Moreover, the metrics for each testing group can be dumped into a JSON file, so that they are more easily accessible to be read and manipulated by automated scripts capable of reading such file format.

The metrics reported in Section 3.5 are already implemented in Revelio.

```yaml
experiment:
    batch_size: 64
    model:
        name: feature_inception_resnet
        args:
            pretrained: true
            feature_name: wavelets
            input_depth: 23
    training:
        enabled: true
        args:
            epochs: 50
            optimizer:
                name: SGD
                args:
                    lr: 0.0005
            loss:
                name: BCEWithLogitsLoss
            callbacks:
              - name: Tensorboard
                args:
                    log_dir: /directory/to/logs
              - ...
```

Listing 2.5: Configuration of the *Model Training* module dedicated to S-MAD training. As reported, is possible to define a specific model, in combination with all the training details, such as the optimizer and callbacks to log the training details.

```yaml
scores:
    bona_fide: /dir/to/{group}_bona_fide_scores.txt
    morphed: /dir/to/{group}_morphed_scores.txt
    metrics: /dir/to/metrics_scores.json
metrics:
  - name: equal_error_rate
  - name: bpcer_at_apcer
    args:
        thresholds:
          - 0.1
          - 0.05
          - 0.01
```

Listing 2.6: Configuration of the *Evaluation* module, in which it is possible to define the metrics output by Revelio framework (in this example: Equal Error Rate (EER) and lowest BPCER related to APCER $\leq 10\%$, APCER $\leq 5\%$ and APCER $\leq 1\%$.

Chapter 3

Datasets and evaluation

In this Chapter we report all the datasets used for training and evaluating both S-MAD (Chapter 5) and D-MAD (Chapter 6) algorithms; these datasets are either publicly available or can be generated by applying public morphing algorithms on face images contained in the original public datasets. In addition, we release[1] the subject pairs used to create the morphed images and the list of the data exploited for training.

As a general overview, some statistics about each dataset are reported in Tables 3.1 and 3.2. In order to facilitate a comparison between the quality of the various morphing algorithms, some visual samples are depicted both in each dataset's corresponding Subsection and in Figure 3.1. A detailed analysis of the composition and data sources of each dataset is reported in the following Subsections.

3.1 Literature datasets

Being morphing attack detection a relatively recent research topic, the number of available datasets in the literature is fairly limited. Moreover, privacy issues prevent the publication and diffusion of standard datasets that can be exploited for training and evaluating MAD algorithms, thus hampering the proliferation of datasets that are specifically created for tackling this problem. To work around these limitations, researchers internally create new datasets starting from existing face-based datasets not specifically constructed for face morphing, including but not limited to the AR [45], FRGC [53], Color Feret [54], LFC-MFD [56], Multimodal BioSecure Database (BMDB) [51], and CelebA [41] datasets. In particular, only faces that are ISO/ICAO-compliant [83] can be exploited to test face recognition and morphing systems.

[1] `https://miatbiolab.csr.unibo.it/revelio-framework`

(a) UBO [6] (b) OpenCV [44] (c) FaceMorpher [55] (d) StyleGAN [36] (e) WebMorph [16]

(f) AMSL [50] (g) Sqirlz Morph [84] (h) FaceFusion [22] (i) UTW [59] (j) NTNU [59]

Figure 3.1: Visual samples of different morphing algorithms used for the training and the experimental evaluation. As shown, the overall quality of morphed images strongly depends on the type of algorithm exploited and may include, among the others, visible artifacts in the background (*e.g.* OpenCV, FaceMorpher, and WebMorph) or the face (*e.g.* UBO, AMSL). It is important to note that morphed images produced through the Sqirlz Morph algorithm are manually retouched.

Morphing Algorithm	**Dataset**	**Data Source**	**#Morphed**	**Quality**
UBO [6]	PMDB	AR - FRGC - Feret	711 - 199 - 198	Medium
OpenCV [44]	Idiap Morph	FRGC - FRLL - Feret	964 - 1222 - 529	Low
FaceMorpher [55]	Idiap Morph	FRGC - FRLL - Feret	964 - 1222 - 529	Low
StyleGAN [36]	Idiap Morph	FRGC - FRLL - Feret	964 - 1222 - 529	Low
WebMorph [16]	Idiap Morph	FRLL	1221	Low
AMSL [50]	Idiap Morph	FRLL	2175	Low
Sqirlz Morph$_D$ [84]	MorphDB$_D$	FRGC - Feret	50 - 50	High
Sqirlz Morph$_{P\&S}$ [84]	MorphDB$_{P\&S}$	FRGC - Feret	50 - 50	High
FaceFusion [22]	ChiMo - FEI	CFD - FEI	8310 - 2000	Medium
UTW [59]	ChiMo - FEI	CFD - FEI	8310 - 2000	Medium
NTNU [59]	ChiMo - FEI	CFD - FEI	8310 - 2000	Medium

Table 3.1: Morphing algorithms and datasets. For each morphing algorithm, the related dataset name, the original source of the images used for the morphing procedure, and the number of morphed images for every data source are reported. The last column reports the quality of morphed images, as discussed in Sections 3.1 and 3.2.

Data Source	#Morphed	#Bonafide	Notes
FRGC [53]	3092	2581	50 P&S
AR [45]	711	1422	-
Feret [54]	1985	2720	50 P&S
FRLL [16]	7062	92	-
CFD [43]	24930	831	-
FEI [79]	6000	200	-

Table 3.2: Analysis of the amount of morphed and bonafide images in relation to each source dataset (see Table 3.1). As shown, the morphed images represent the large majority of available data during the training and testing phases.

3.1.1 Progressive Morphing Database (PMDB)

This dataset [26] is built starting from three well-known datasets, *i.e.* AR [45], FRGC [53] and Color Feret [54], using the public morphing algorithm described in [26], creating a total of 1108 morphed images, a sample of which can be found in Figure 3.2. 280 subjects were used to generate the morphed images, split into 134 males and 146 females.

It is important to note that on PMDB no manual retouching procedures are applied to morphed images in order to enhance the visual quality; therefore, the images may contain artifacts (such as blurred areas or ghosts). The background is automatically replaced by the morphing algorithm, then it does not include any artifacts.

Figure 3.2: Visual samples of three images from the PMDB [26] dataset.

3.1.2 Idiap Morph

This dataset [62, 61] is a publicly available set of several datasets, specifically consisting of five subsets created with five distinct morphing algorithms (OpenCV [44], FaceMorpher [55], StyleGAN [36], WebMorph [16] and AMSL [50]), exploiting the face images belonging to the Feret [54], FRGC [53] and Face Research Lab London Set [16] (in this thesis referred as FRLL) datasets as input data. A sample image from each morphing algorithm used in this dataset can be found in Figure 3.3.

As depicted in Figures 3.1b and 3.1c, the overall visual quality of morphed images created employing the OpenCV and FaceMorpher algorithms is negatively influenced by the heavy presence of artifacts, located both in the background and foreground (*i.e.* the face) of images. In morphed faces generated with the StyleGAN-based approach (as shown in Figure 3.1d), visual artifacts are less visible, but common GAN-related textures are still present and detectable [87]. The AMSL morphing algorithm (a sample of which can be found in Figure 3.1f) is exploited to produce 2175 morphed images starting from 102 adult faces, with a morphing factor equal to 0.5. The interesting feature of this morphing algorithm is represented by the compression procedure applied to all images, in order to fit on the single chip of the eMRTD that is available in official documents: therefore, the available images are encoded using the JPEG Sequential Baseline (ISO/IEC 10918-1) mode of operation [83] and have a maximum size of 15 kB. We observe that the compression procedure tends to make the S-MAD task more challenging since it deletes most of the artifacts possibly introduced by the morphing algorithm.

3.1.3 MorphDB

This dataset [26], built using images belonging to the Color Feret [54] and FRGC [53] datasets, consists of 100 morphed images created starting from 50 male and 50 female subjects using the *Sqirlz Morph 2.1* [84] algorithm. A sample image from this dataset can be found in Figure 3.4.

Unfortunately, this dataset is not publicly released, but it can be found on the FVC-onGoing [5] platform to be used as a test dataset as in the Revelio framework. Despite this issue, it represents an interesting testing dataset, since all morphed images have been manually retouched, and therefore the final visual quality is excellent. This dataset contains also a set consisting of real printed-and-scanned (P&S) images, *i.e.* bonafide and morphed images that have been realistically printed and scanned with professional tools. These two factors combined make this dataset particularly challenging, although the limited number of images may make it unsuitable for conducting an extensive performance review of a MAD algorithm.

(a) OpenCV [44] (b) FaceMorpher [55] (c) StyleGAN [36]

(d) WebMorph [16] (e) AMSL [50]

Figure 3.3: Visual sample of an image for each morphing algorithm used in the Idiap Morph [62, 61] dataset.

(a) Digital (b) Printed & scanned

Figure 3.4: Visual sample of an image from the MorphDB [26] dataset, in both digital and printed-and-scanned form.

3.2 Proposed datasets

To further increase the volume of available training data, we introduce two datasets that can be used for Morphing Attack Detection: *ChiMo* and *FEI*. Both these datasets are based on others that are not explicitly designed for solving Morphing Attack Detection. To generate the morphed images, the same protocol has been employed: for each subject, five other subjects of the same ethnicity and gender have been selected for morphing; in order to maximize the attack potential of the morphed images (*i.e.* the probability of fooling FRSs), the average face verification scores of three commercial SDKs (*VeriLook*[2], *Cognitec*[3] and *Innovatrics*[4]) have been used to select the most similar subjects for each individual. Then, two morphing factors (0.3 and 0.5) and three different morphing algorithms (FaceFusion [22], UTW [59] and NTNU [59]) are applied for each subject pair.

3.2.1 ChiMo

The *ChiMo* dataset has been generated using the images (with neutral expression) of the *Chicago Faces Database* (CFD) [43] that includes images of 831 subjects of varying ethnicities.

By implementing the protocol depicted above, this dataset contains 24930 morphed images (8310 per morphing algorithm). A sample image from each morphing algorithm used in this dataset can be found in Figure 3.5.

Finally, two versions of this dataset are created: the first one contains the digital images as produced through the morphing procedure, while in the second (referred to with the subscript *JPG*) we applied a compression procedure similar to the one applied on AMSL, thus obtaining images with a maximum size of 15 kB.

3.2.2 FEI

The *FEI* dataset has been generated using the images contained in the *FEI Face Database* [79], which includes 200 subjects, equally split between male and female.

By implementing the protocol depicted above, this dataset contains 6000 morphed images (2000 per morphing algorithm). A sample image from each morphing algorithm used in this dataset can be found in Figure 3.6.

For each subject two frontal images (one with a neutral or non-smiling expression and the other with a smiling facial expression) are available, thus making this dataset suitable for D-MAD as well. All faces are mainly represented by subjects between 19 and 40 years old with distinct appearances, hairstyles, and accessories.

[2] `www.neurotechnology.com/verilook.html`

[3] `www.cognitec.com`

[4] `www.innovatrics.com/`

(a) FaceFusion [22] (b) UTW [59] (c) NTNU [59]

Figure 3.5: Visual sample of an image for each morphing algorithm used in the ChiMo dataset.

(a) FaceFusion [22] (b) UTW [59] (c) NTNU [59]

Figure 3.6: Visual sample of an image for each morphing algorithm used in the FEI dataset.

3.3 D-MAD suitability

With the exception of ChiMo, all datasets are also suitable for D-MAD: indeed, each subject can be paired with another picture of the same subject with a different pose and possibly lighting conditions. This condition is necessary for a dataset to be suitable for D-MAD because it would be incorrect to use the same image both to generate the morphed picture and as a live image.

The statistics about the datasets used for the D-MAD task can be found in Table 3.3.

Morphing Algorithm	**Dataset**	**#Bona fide**	**#Morphed$_C$**	**#Morphed$_A$**
UBO [6]	PMDB	280	1108	1108
OpenCV [44]	Idiap Morph	3951	5021	5021
FaceMorpher [55]	Idiap Morph	3951	4323	5022
StyleGAN [36]	Idiap Morph	3951	4323	5022
Sqirlz Morph$_D$ [84]	MorphDB$_D$	756	396	360
FaceFusion [22]	FEI	400	4000	4000
UTW [59]	FEI	400	4000	4000
NTNU [59]	FEI	400	4000	4000

Table 3.3: Morphing algorithms and datasets. For each morphing algorithm, the related dataset name, the original source of the images used for the morphing procedure, and the number of morphed images for every data source are reported. The last column reports the quality of morphed images, as discussed in Sections 3.1 and 3.2.

3.4 Real-world applicability

One significant issue that afflicts many of the presented datasets is that they fail to accurately represent the different quality levels of the pictures used as input.

Indeed, the majority of the images used for training MAD algorithms have high definition and low compression ratios; however, while these conditions are ideal, they are not realistic for real-world use cases.

Two commonplace factors that rapidly deteriorate the quality of the input images can be found:

- *strong lossy compression*: the subject's mugshot image stored into the chip in compressed form, either using the JPEG Sequential Baseline (ISO/IEC 10918-1) mode of operation or the JPEG-2000 Part-1 Code Stream Format (ISO/IEC 15444-1) [83]; considering the minimum image size requirement of 11 kB given in [33], most of the issuing authorities adopt a compressed image size of around 12-15 kB;

- *printing and scanning*: in many countries, the digital photo acquired by professional photographers is printed on paper and then scanned by the officer to be included in the document during the eMRTD issuing process.

Both these processes, which can be seen in Figure 3.7, have the effect of concealing many of the artifacts left by morphing algorithms, thus hampering the performance of MAD systems that have been trained only on high-quality images.

(a) Digital (b) Printed & scanned

(c) Compressed (d) P&S and compressed

Figure 3.7: Visual sample of the same image with different artifacts. As shown, strong image compression and P&S severely degrade the quality of the image (d) with respect to the original digital version (a).

3.5 Literature metrics

In order to evaluate and compare MAD systems, there are several metrics commonly used for performance assessment in the context of morphing detection [70]:

- *Bona Fide Presentation Classification Error Rate* (BPCER): represents the proportion of bona fide images wrongly classified as morphed; if the morphed class is the positive one, this metric is equivalent to the *False Rejection Rate* (FRR). It can be mathematically defined as follows:

$$\text{BPCER}(\tau) = \frac{1}{N} \sum_{i=1}^{N} H(b_i - \tau) \tag{3.1}$$

- *Attack Presentation Classification Error Rate* (APCER): represents the proportion of morphed images wrongly accepted as bona fide; if the morphed class is the positive one, this metric is equivalent to the *False Acceptance Rate* (FAR). It can be mathematically defined as follows:

$$\text{APCER}(\tau) = 1 - \left[\frac{1}{M} \sum_{i=1}^{M} H(m_i - \tau) \right] \tag{3.2}$$

In both definitions, τ is the score threshold on which b_i, m_i, the detection scores, are compared; $H(x) = \{1 \text{ if } x > 0, 0 \text{ otherwise}\}$ is defined as a step function.

Typically, the BPCER is measured 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. Ideally, a MAD algorithm employed in a real-world setting would need to operate at a low APCER (*i.e.* letting almost no criminals through) of around 0.1%, while maintaining an acceptable corresponding BPCER (*i.e.* generating few false positives) of around 1%.

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. An example of DET curve can be found in Figure 3.8: the ratio of false negatives increases when moving to the right, while the ratio of false positives increases when moving up. The curve must at least intersect the

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.

3.6 Proposed metrics

The results reported on different datasets, using several performance indicators, can be sometimes dispersive and difficult to analyze as a whole. To the best of our

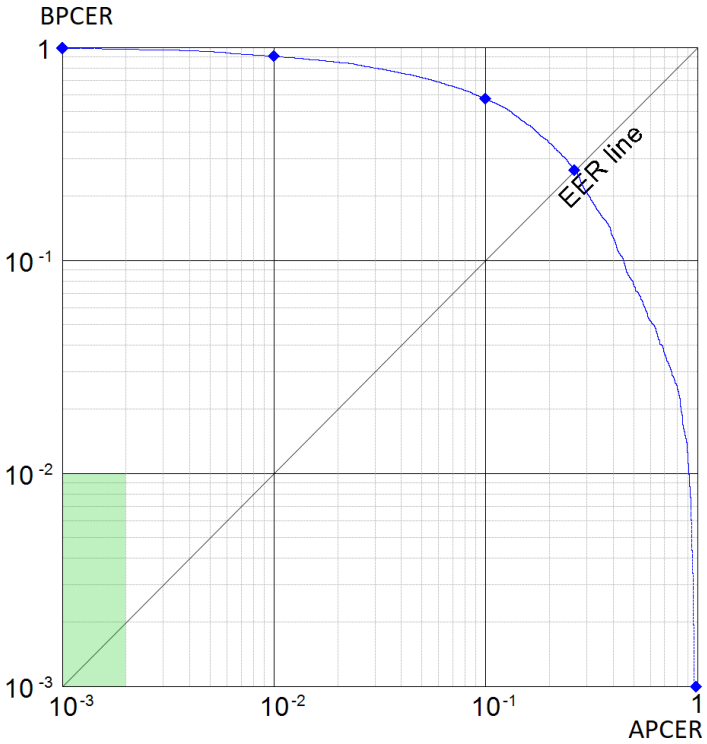


Figure 3.8: An example of a DET curve; the green area in the bottom-left corner represents the ideal operating window for real-world uses.

knowledge, we are not aware of widely-accepted methods capable of mitigating this problem; therefore, we introduce our proposed approach to address this very issue.

3.6.1 Weighted Average Error across Datasets (WAED)

To summarize and simplify the comparison of diverse approaches across different testing datasets, we introduce therefore a new metric, namely *Weighted Average Error across Datasets* (in short, WAED), that aims to condense the aforementioned set of error metrics $\mathcal{E}$ computed on different testing datasets $\mathcal{D}$ into a single value:

$$\text{WAED} = \sum_{E \in \mathcal{E}} \sum_{D \in \mathcal{D}} w_D w_E E(D) \tag{3.3}$$

where:

- $E(D)$ is the value of the error indicator $E \in \mathcal{E}$, measured on the dataset $D \in \mathcal{D}$;
- w_E is a weighting factor assigned to each error indicator in order to focus our attention on the error indicators which are more relevant for a real-

world scenario (*e.g.* $B_{0.01}$). The weights considered for the WAED metric computation (see Table 3.4) are chosen by assigning the majority of the weight to the most common real-world operating point (*i.e.* $B_{0.01}$), followed by the EER, as it is useful for evaluating the performance of the system at a glance, and finally the other two chosen operating points (*i.e.* $B_{0.05}$ and $B_{0.1}$);

- w_D is a dataset weight that empirically measures the dataset complexity: a good method to quantify this factor is by measuring the similarity of the morphed images to the two contributing subjects. In particular, for each dataset $D \in \mathcal{D}$, we compute the value s_D, through the comparison of each morphed image $m_i \in D$ with the S bona fide images $b_{i,j}$ used in the morphing process. The comparison is done on K different commercial face verification SDKs, as follows:

$$s_D = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{S}\sum_{j=1}^{S}\frac{1}{K}\sum_{k=1}^{K}\frac{s_k\left(m_i, b_{i,j}\right) - thr_k}{thr_k} \tag{3.4}$$

  where $M = |D|$, $S = 2$ since we tackle images produced through two-subjects morphing algorithms, and $K = 3$ since we exploit Verilook, Cognitec, and Innovatrics SDKs, respectively. To make comparable the scores of different SDKs, the similarity score $s_k(m_i, b_{i,j})$ provided by each SDK is normalized according to the FAR_{1000} threshold (thr_k) provided by the SDK.

  Finally, the single dataset scores are normalized in the range $[0, 1]$ as follows:

$$w_D = \frac{s_D}{\max\limits_{D\in\mathcal{D}} s_D} \tag{3.5}$$

The proposed metric produces a single numeric value in the range $[0, 1]$ with which comparisons are simplified: being an overall error measure, low values are desirable.

Metric	**Weight** (w_E)
EER	.30
$\mathbf{B}_{0.1}$	.10
$\mathbf{B}_{0.05}$	.20
$\mathbf{B}_{0.01}$	.40

Table 3.4: Metric weights (w_E) used for the proposed WAED metric.

3.7 Benchmarks

As mentioned in Section 3.1, many MAD systems are trained on datasets that are internally created, which often lack diversity with regard to size, image quality, realistic post-processing, and variability of morphing algorithms. These issues become particularly apparent when evaluating different MAD systems.

To have a fair comparison between different MAD algorithms, and to measure their performance in conditions as close as the ones found in the real world, two standard benchmarks have been proposed: the *SOTAMD* dataset (available through the FVC-onGoing[5] [5] platform) and the *NIST FRVT MORPH*[6] [48] platform.

Both of these are more generally known as *sequestered datasets*: researchers do not have access to the images contained in these benchmarks, so they cannot be used to train the models or guide in any way the choice of the best hyperparameters.

3.7.1 SOTAMD sequestered test set

To have a complete and fair performance comparison between MAD systems, a new dataset is created as a joint effort in an EU-funded project, known as *State-Of-The-Art Morphing Detection* (SOTAMD). The SOTAMD dataset consists of the following:

- *Enrollment images*: bona fide face images meeting the requirements of passport application photo capture (*e.g.* photographer studio);
- *Gate images*: bona fide face images captured live with a face capture system in an Automated Border Control (ABC) gate;
- *Chip images*: compressed face images stored on an electronic Machine-Readable Travel Document (eMRTD);
- *Morphed face images*: morphed images created from the pool of enrolled face images; this database contains both digital (with and without postprocessing) and printed and scanned images.

The final number of images is 300 bona fide, 2045 morphed, and 1500 at the gate (1096 and 3703 for bona fide and morphed in the P&S scenario, respectively). There are 75 unique pairs of candidates for morphing from 150 individuals of various ethnicities and ages.

[5] `https://biolab.csr.unibo.it/fvcongoing`
[6] `https://pages.nist.gov/frvt/html/frvt_morph.html`

3.7.2 NIST Face Recognition Vendor Test

The *NIST Face Recognition Vendor Test*, also known as NIST FRVT MORPH, provides ongoing independent testing of MAD systems. The test leverages several datasets created using different morphing techniques in order to evaluate the robustness of the morphing attack detector. The test datasets are divided into three tiers, with increasing difficulty levels.

- *Tier 1*: lower quality morphs created with readily accessible tools available to non-experts, such as online tools from public websites and free mobile applications. These morphs are created using low-effort processes and are generally low quality and contain large amounts of morphing artifacts that are visible to the human eye.
- *Tier 2*: morphs generated using automated morphing methods based on academic research and best practices. Automated methods allow for the generation of morphs in large quantities for testing.
- *Tier 3*: higher quality morphs created using commercial-grade tools with manual processes. These are high-quality morphs with very minimal visible morphing artifacts.

Chapter 4

Related works

This Chapter contains a non-exhaustive selection of relevant proposed methods to tackle both S-MAD and D-MAD problems.

4.1 Single-image Morphing Attack Detection

4.1.1 Detection of morphed faces from single images: a multi-algorithm fusion approach

In [64], summarized in Figure 4.1, the authors propose a system employing traditional methods for feature extraction and relying on several *Support Vector Machines* [14] (SVMs) to produce intermediate *normalized attack detection scores* in the $[0, 1]$ range, which are then combined through the sum-rule with proper normalization [34].

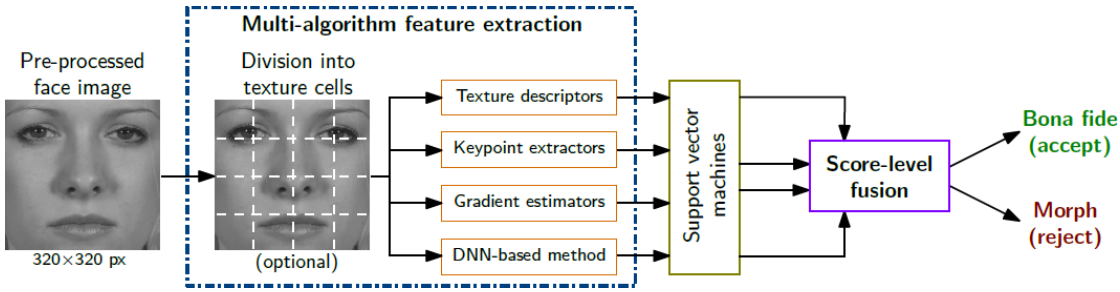


Figure 4.1: Overview of the S-MAD method proposed in [64], based on multiple *Support Vector Machines* (SVMs) [14] that are trained on different features, and whose scores are combined using a normalized sum-rule [34]. Image source [64].

The employed features can be categorized into four disjoint groups:

- *Texture descriptors*: *Local Binary Patterns* (LBP) [40] and *Binarized Statistical Image Features* (BSIF) [35] are used to capture any possible texture differences between bona fide and morphed images.

- *Keypoint extractors*: *Scale Invariant Feature Transform* (SIFT) [42] and *Speeded Up Robust Features* (SURF) [4] are employed because, according to the authors, morphed images tend to present less key locations; therefore, the number of detected keypoints is used as a discriminating feature.

- *Gradient estimators*: *Histogram of Oriented Gradients* (HOG) [73] and sharpness (*i.e.* the mean of the gradient in two dimensions) are adopted because the morphing process reduces the steepness of such gradients.

- *Deep learning-based methods*: *OpenFace* [3] is used thanks to the advancements in face recognition.

A commercial off-the-shelf face recognition system is used to evaluate the performance of the proposed algorithm. Results show that LBP represents the best solution as a single descriptor, followed by BSIF, SURF, SIFT, sharpness, and HOG. Moreover, the fusion of different features provides better results, but with limited absolute improvement. The best combination consists in combining the LBP, SIFT, and sharpness features.

4.1.2 Detection of face morphing attacks based on PRNU analysis

In [68], the authors propose the analysis of the camera's sensor noise, known as *Photo Response Non-Uniformity* (PRNU) [12], to classify images into either the bona fide or morphed classes.

PRNU has previously been utilized as a reliable tool to perform various forensic tasks, including detecting digital forgeries. The PRNU origins from slight variations among individual pixels during the photoelectric conversion in digital image sensors.

The method, summarized in Figure 4.2, consists in extracting the PRNU noise from the image and extracting two kinds of features, namely spatial and spectral features; the former aim at analyzing the distribution of the PRNU noise, while the latter try to reveal any alterations of the PRNU signal caused by the morphing process. Feature aggregation is obtained by sampling the minimum or maximum score among the individual cells, and the final decision is taken using a threshold.

Experimental results show good accuracy with the respect to traditional image descriptors and deep features from FaceNet [71], even though the results are strongly affected by the exploited morphing algorithm.

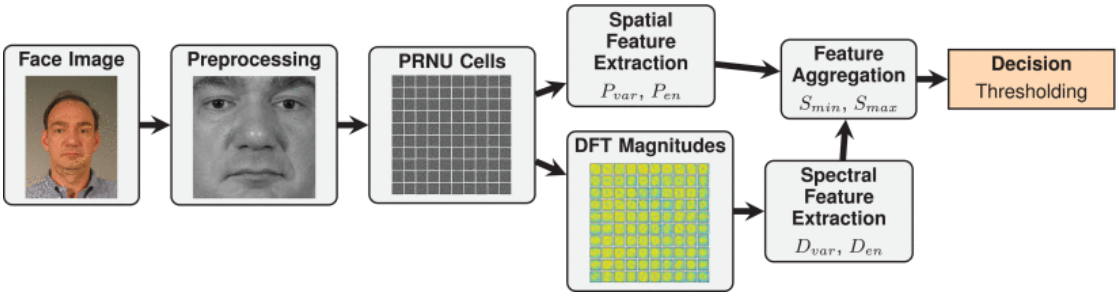


Figure 4.2: Overview of the S-MAD method proposed in [68], based on the extraction and analysis of the *Photo Response Non-Uniformity* (PRNU) [12] noise. Image source: [68].

4.1.3 Face morphing detection in the presence of printing/scanning and heterogeneous image sources

In [24], the authors propose to use well-known networks (in particular AlexNet [39], VGG-19 [74], VGG-Face2 [9] and VGG-Face16 [52]) to determine whether an image is morphed or not.

Due to the scarcity of training data, a first fine-tuning step is performed on digital images starting from networks pre-trained either on the Imagenet [17] or the VGG-Face2 [9] datasets. Then, a second step of fine-tuning is conducted to improve the performance of the algorithm on printed-and-scanned images. Indeed, results show that networks trained only on digital images are not able to effectively classify images that have undergone a printing and scanning process. The exploiting of simulated printed-and-scanned images during the model's training allows, in some cases, for a significant improvement.

4.1.4 Morphing detection based on regional analysis of local frequency content

In [47], the author explores the possibility of classifying images by analyzing them in the frequency domain.

The author employs an SVM and a DNN for classifying the spectrums obtained after applying a 2D Discrete Fourier Transform and computing the power spectrum on the images.

Results show that, in a single-image setting, the SVM has a slightly higher classification accuracy than the DNN. However, being the study limited in scope, it is difficult to compare the obtained results to the others that can be found in the literature.

4.1.5 Attention aware wavelet-based detection of morphed face images

In [1], a summary of which can be found in Figure 4.3, the authors propose the use of uniform wavelet decompositions to extract forensic features from the images and then classifying them by employing an Inception-Resnet V1 [77] with three custom attention modules.

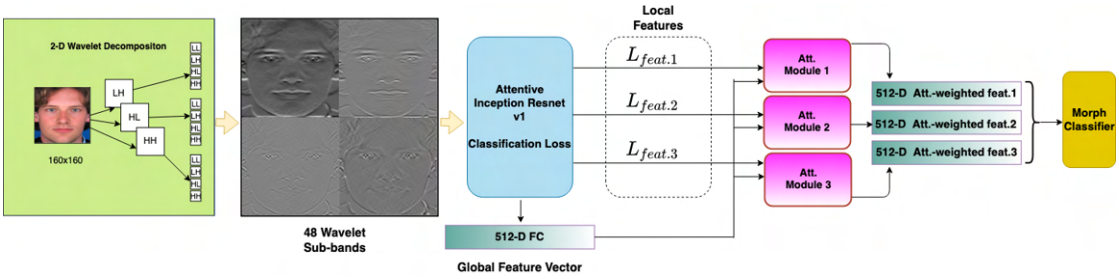


Figure 4.3: Overview of the S-MAD method proposed in [1], based on uniform wavelet decompositions for extracting forensic features, to be classified using an Inception-Resnet V1 [77]. Image source: [1].

The authors affirm that most artifacts produced by facial image morphing techniques lie within the high-frequency spectrum, and therefore using wavelet decomposition allows them to select the desired wavelet sub-bands by discarding the low-frequency ones. In particular, the authors apply a three-level undecimated 2D wavelet decomposition on the input image, discarding the LL sub-band after the first decomposition. In total, each image is decomposed into 48 sub-bands that are then stacked channel-wise. Then, the resulting volume is fed into an Inception-Resnet V1 network, which has been adapted to support a 48-channel input volume. Moreover, the authors insert three attention modules: each module receives as input the global feature vector and the activations of a specific volume inside the network, to produce an attention-weighted local feature vector. According to the authors, the three attention modules can emphasize the artifacts stemming from the morphing attack, leading to more accurate detection of morphed images. The resulting three attention-weighted local feature vectors are then concatenated and used as input for the classifier.

Results show that while the conducted ablation study proves that the inserted attention modules are effective in improving the model's performance, the authors also demonstrate that the use of undecimated wavelet decomposition is actually advantageous only in limited cases, while in the others a classifier that takes RGB images and uses an Inception-Resnet V1 as backbone provides generally lower error rates.

4.2 Differential Morphing Attack Detection

4.2.1 Detecting morphed face images using facial landmarks

In [67], summarized in Figure 4.4, the authors propose an algorithm developed on the basis that facial landmarks tend to be averaged between two subjects during the creation of morphed images. In particular, they observe that the distance of a specified landmark between two bona fide images of the same individual is likely to be smaller than the distance between the same landmark from a genuine image of the subject and a morphed image of another one. Following these considerations, the authors extract two types of features: Euclidean distances between landmarks, and angles between a pre-defined set of neighboring landmarks.

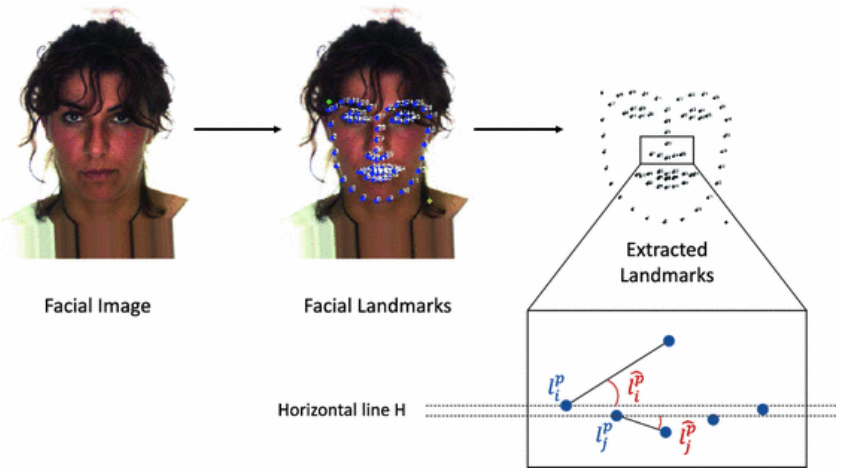


Figure 4.4: Overview of the D-MAD method proposed in [67], based on comparing the Euclidean distance of facial landmarks between the suspected morphed and live images. Image source: [67].

Results show that the best performance can be achieved using an SVM with a *Radial Basis Function* (RBF) [10] kernel. Nevertheless, though results indicate that some information about the morphing process can be derived from facial landmarks, the presented solution does not possess sufficient overall performance to be viable for practical applications.

4.2.2 Face demorphing

In [25], authors explore the idea of trying to reverse the morphing process and extracting the identity of the legitimate document owner. The method, whose pipeline can be outlined in Figure 4.5, employs a face recognition system to compare the live and the potentially morphed images; then, the same system compares the

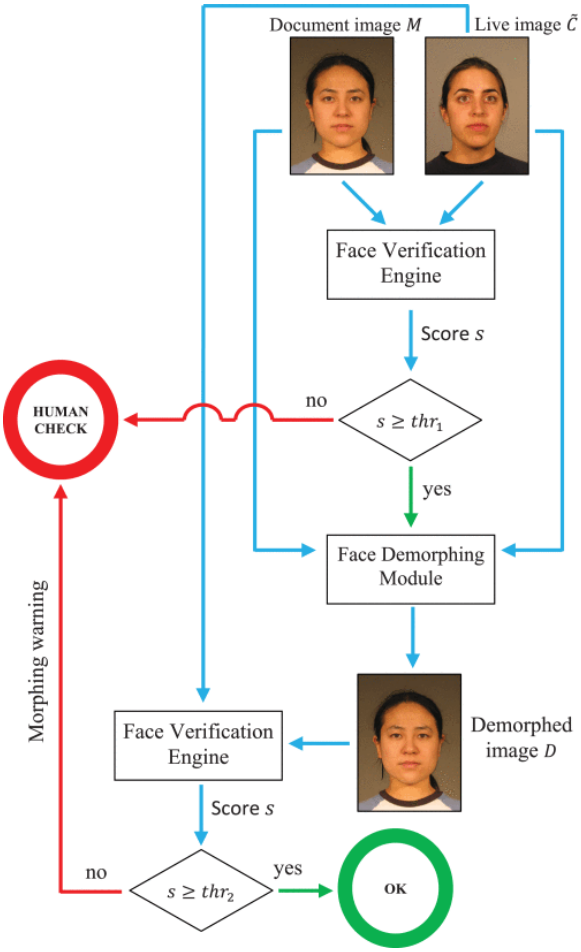


Figure 4.5: Overview of the D-MAD method proposed in [25], based on reversing the morphing process to extract the identity of the legitimate document owner. Image source: [25].

live image and the picture obtained via the demorphing process. Only if both checks succeed, then the document is considered valid and the ABC gate is allowed to open. On the contrary, if any face verification check fails, an alert is sent to a human officer. In particular, if the second check is unsuccessful, a warning of a possible morphing attempt is triggered.

Results are collected using the VeriLook FRS and suggest that images morphed with a factor $\alpha \in [0.2, 0.3]$ represent the best trade-off between the probability of the morphed image being accepted by both face verification systems and human officers. Moreover, the demorphing process is able to reduce the chance (according to the authors, from 60–70% to 6–10%) of a criminal fooling an ABC gate, while at the same time maintaining the number of false positives relatively limited.

4.2.3 Deep face representations for differential Morphing Attack Detection

In [66], authors propose to exploit a pre-trained deep neural network, *i.e.* FaceNet [71] and ArcFace [18], for feature extraction.

As these networks are pre-trained on datasets with no morphed images, the extracted features do not contain any information specific to any morphing algorithm. The algorithm, depicted in Figure 4.6, extracts the embeddings for both the live and the suspected morphed images and combines them by subtracting them. Then, an SVM with an RBF kernel is trained on these difference vectors and a score is produced.

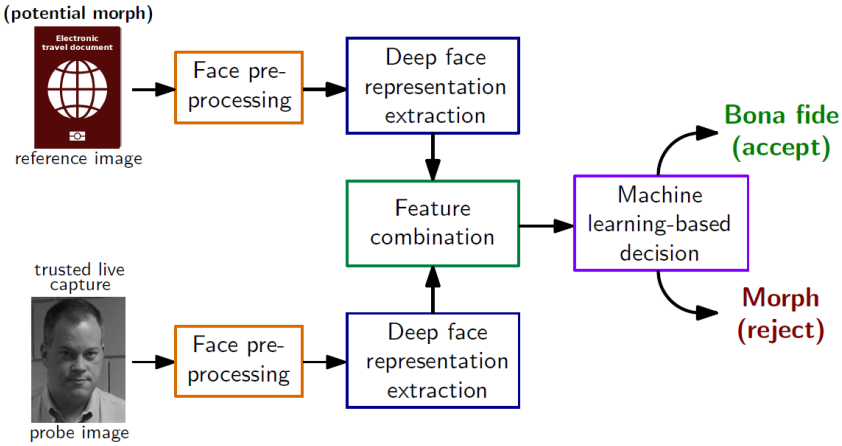
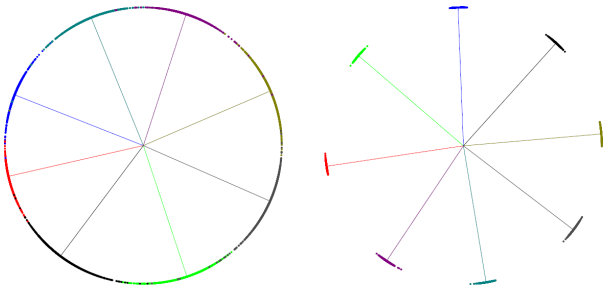


Figure 4.6: Overview of the D-MAD method proposed in [66], based on combining and then classifying the features emitted by a state-of-the-art backbone such as ArcFace [18]. Image source: [66].



(a) Softmax (b) ArcFace

Figure 4.7: Comparison of 2D features of 8 identities (one per color) produced using the softmax and ArcFace losses. Dots indicate samples and lines represent the center direction of each identity. Image source: [18].

Results show the effectiveness and robustness of the proposed algorithm. Authors show that the ArcFace backbone proves to be very effective at producing robust embeddings, also thanks to its specific loss function [18] that maximizes the geodesic distance between different identities, as shown in Figure 4.7. Moreover, this D-MAD algorithm provides remarkable results against the SOTAMD sequestered test set, obtaining a D-EER of 4.54%.

However, as the underlying feature extractor does not have any knowledge about face morphing, the algorithm proposed by the authors can be essentially considered a face verification system rather than a morphing attack detector.

Chapter 5

Single-image MAD experiments

In this Chapter, we define the training and testing protocols used in our experimental evaluation for S-MAD systems, describing the datasets used and the training and testing split. In particular, we aim to clearly define a common protocol for future MAD proposals, seizing the opportunity to propose a comprehensive empirical comparison of different MAD approaches available in the literature.

5.1 Datasets and protocols

In Revelio, and in all the following experiments and tables, we group the train and test datasets relying on the morphing algorithm used to produce morphed images, as also reported in the first column of Table 3.1. When training and evaluating the S-MAD experiments the images from the FEI data source are not included, as it is a dataset that was introduced after the end of the experiments. Therefore, the FaceFusion, UTW and NTNU morphing algorithms are comprised exclusively of images from the ChiMo dataset (8310 morphed images).

We believe this data organization is useful to analyze the MAD performance in relation to different morphing algorithms that represent a key element in the development of MAD techniques [70]. Then, different datasets can be grouped in the same set; for instance, the StyleGAN-based [36] morphing algorithm groups the subsets belonging to the Idiap Morph built on three different data sources (FRGC, FRLL and Color Feret).

All the experiments have been carried out following the dataset organization and training/testing protocols described in Table 5.1. Specifically, we create a challenging setup following these considerations:

- Morphing algorithms used to produce images in training and testing splits are different; more precisely, we create one validation and one test set: the first one is a subset (20% of the images) taken from the same datasets used

Morphing Algorithm	Train/Val (%)	Test (%)
UBO [6]	70+10	20
OpenCV [44]	70+10	20
FaceMorpher [55]	70+10	20
StyleGAN [36]	70+10	20
WebMorph [16]	0	100
AMSL [50]	0	100
Sqirlz Morph$_D$ [84]	0	100
Sqirlz Morph$_{P\&S}$ [84]	0	100
FaceFusion [22]	0	100
UTW [59]	0	100
NTNU [59]	0	100

Table 5.1: Morphing algorithms and datasets in the *Revelio* framework used for the experimental evaluation. For each morphing algorithm, the percentage of images available during the training, validation, and testing phases are shown.

for model training so that the morphing algorithms coincide with those in the training set, while the second one, on which the WAED metric is computed (see Section 3.6.1), contains all the morphed images generated with unseen morphing algorithms.

- The training datasets contain morphed images with low visual quality due to, for instance, the presence of artifacts, as shown in Figure 3.1, while the test set only includes medium or high-quality morphed images, due to the human intervention in retouching procedures (MorphDB) or the absence of visible artifacts in the backgrounds (ChiMo).

It is worth noting that this setting assures a demanding cross-morphing algorithm evaluation, aiming to verify the generalization capabilities of the investigated MAD methods. Besides, all the images taken from the Chicago Face Dataset belong to subjects never seen during the training procedure.

Table 3.2 reports, for each public face dataset, the number of bona fide images considered in the experiments and the total number of morphed images derived from that dataset. It is important to note the unbalanced amount of bona fide and morphed images, which contributes in making challenging the proposed setting.

In order to evaluate and compare the investigated MAD methods, we use the metrics reported in Sections 3.5 and 3.6.

Moreover, as reported in Section 3.6.1, we employ the proposed WAED metric to summarize and simplify the comparison of the diverse approaches across the different testing datasets. The weights for the testing datasets employed in the

S-MAD experiments (w_D) are computed by following the instructions mentioned in Section 3.6.1, obtaining the results that are reported in Table 5.2.

Dataset	Weight (w_D)
FaceFusion$_{[JPG]}$	1.00
NTNU$_{[JPG]}$	.94
UTW$_{[JPG]}$	.88
Webmorph	.80
Sqirlz	.78
AMSL	.77

Table 5.2: Dataset weights (w_D) used for the proposed WAED metric (see Section 3.6.1). Subscript $[JPG]$ denotes both versions of the dataset, with digital and compressed images.

5.2 Experimental results

As previously mentioned, experimental results are reported grouped by morphing algorithms, and then a single group can refer to more than one dataset. Associations between the original dataset and the morphing algorithms are reported in Table 3.1.

In all the following experiments, all the networks are pre-trained either on the ImageNet [17] or VGG-Face2 [9] datasets (weights downloaded from the official PyTorch[1] storage), and trained using the binary cross-entropy loss function. As an optimizer, we use the *Stochastic Gradient Descent* (SGD), with a learning rate in the range of $[10^{-3}, 5 \cdot 10^{-3}]$ and early-stopping (with patience of 5 epochs and a minimum improvement of 10^{-3}) to prevent overfitting computed on the validation set. All the configuration settings and trained models are publicly released[2].

5.2.1 Investigation on face detectors

Several robust face detection techniques are available in the literature and this first set of experiments aims to compare the most promising ones and to evaluate their impact on S-MAD performance, being aware that in this application scenario face detection is quite a simple task, since all input images are fully ISO/ICAO-compliant [83], with natural expression, acquired in constrained (frontal) pose and lighting conditions, etc.

Then, we focus on testing three different face detectors widely used in the literature, in particular in MAD methods, based on Machine and Deep Learning

[1] `https://pytorch.org`

[2] `https://miatbiolab.csr.unibo.it/revelio-framework/`

Morphing Alg.	DLib [37]				OpenCV [81]				MTCNN [86]			
	EER	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
UBO	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	.014	**.000**	**.000**	.014
OpenCV	**.001**	**.000**	**.000**	**.001**	.005	**.000**	**.000**	.002	.018	.002	.006	.035
FaceMorpher	**.002**	**.000**	**.000**	**.000**	.002	**.000**	**.000**	.002	.007	.001	.002	.006
StyleGAN	**.001**	**.000**	**.000**	**.000**	.003	**.000**	**.000**	.001	.018	.002	.004	.021
AMSL	.302	.650	.700	.950	.300	.700	.750	.900	**.112**	**.150**	**.300**	**.650**
Webmorph	.400	**.700**	.900	1.000	.400	.750	**.750**	.900	**.362**	.800	.800	**.850**
Sqirlz Morph_D	.041	.012	.033	.122	**.032**	**.000**	**.008**	.081	**.032**	.021	.033	**.062**
FaceFusion	**.255**	.656	.803	.945	.277	**.598**	**.732**	**.898**	.300	.759	.876	.970
NTNU	.215	.680	.854	.978	.249	.651	.826	.966	**.182**	**.489**	**.752**	**.954**
UTW	.538	.912	.959	.992	.516	.899	.946	.996	**.300**	**.567**	**.714**	**.878**
FaceFusion_{JPG}	.445	.832	.922	.981	.450	.859	.924	.983	**.270**	**.503**	**.663**	**.886**
NTNU_{JPG}	.491	.901	.946	.990	.471	.894	.948	.987	**.327**	**.659**	**.786**	**.931**
UTW_{JPG}	.491	.865	.929	.982	.508	.878	.929	.983	**.363**	**.709**	**.806**	**.931**
WAED $\downarrow$	.6944				.6800				**.5831**			

Table 5.3: Morphing detection scores across different Face Detectors given a fixed ResNet-50 [32] detector. Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq$ 10%, $\leq$ 5%, and $\leq$ 1%, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

techniques: DLib [37], Haar cascades-based [81] (here referred as OpenCV) and MTCNN [86]. Experiments are carried out in combination with a *ResNet-50* [32] architecture, whose effectiveness has been widely documented in the literature for several classification tasks, including MAD [7, 26].

The results reported in Table 5.3 suggest that the MTCNN face detector leads to the best accuracy, while DLib and OpenCV have similar lower values. As depicted in Figure 2.2 the face crop provided by MTCNN detector includes a wider facial area and then tends to include facial parts (chin and outline) in which the morphing procedure usually leaves artifacts.

As mentioned, the choice of the best algorithm and the computation of the WAED metric is based on the results obtained on the second group of testing datasets, in which morphed images have been created with morphing algorithms different from the ones used for the training images.

5.2.2 Investigation on DNN architectures

In the second part of the experiments, we aim to define the best architecture to tackle the morphing classification task.

In [27], authors proposed to exploit well-known deep architectures, ranging from AlexNet [39] to VGG-Face [74], to address the S-MAD task. Reported results seem to suggest that a deep learning approach can achieve high accuracy, provided that

Morphing Alg.	ResNet-50 [32]				Inception-Resnet V1 [77]				Vision Transformer [20]			
	EER	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
UBO	.014	**.000**	**.000**	.014	**.003**	**.000**	**.000**	**.000**	.006	**.000**	**.000**	.002
OpenCV	.017	.002	.006	.022	**.008**	**.000**	**.000**	**.006**	.014	**.000**	.004	.039
FaceMorpher	.006	.001	.002	.005	**.004**	**.000**	**.000**	**.003**	.006	**.000**	**.000**	.005
StyleGAN	.017	.002	.003	.018	.013	**.000**	**.000**	.021	**.009**	**.000**	.002	**.009**
AMSL	.112	.150	.300	.650	**.005**	**.000**	**.000**	**.000**	.150	.250	.300	.350
Webmorph	.362	.800	.800	**.850**	**.250**	**.350**	**.650**	.900	.300	.400	**.650**	**.850**
Sqirlz Morph_D	.032	.021	.033	**.062**	**.024**	**.012**	**.025**	.123	.054	.015	.058	.118
FaceFusion	.300	.759	.876	.970	**.114**	**.136**	**.243**	**.502**	.240	.440	.579	.829
NTNU	.182	.489	.752	.954	**.114**	**.132**	**.261**	**.519**	.228	.448	.598	.897
UTW	**.300**	**.567**	**.714**	**.878**	.312	.641	.769	.942	.439	.750	.836	.951
FaceFusion_{JPG}	.270	.503	.663	.886	**.125**	**.165**	**.351**	**.669**	.186	.278	.391	.679
NTNU_{JPG}	.327	.659	.786	.931	**.158**	**.265**	**.448**	**.703**	.223	.369	.507	.769
UTW_{JPG}	.363	.709	.806	.931	**.315**	**.643**	**.753**	**.918**	.324	.709	.826	.959
WAED $\downarrow$	.5831				**.3915**				.5103			

Table 5.4: Morphing detection scores across different architectures given a fixed Face Detector (MTCNN). Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq 10\%$, $\leq 5\%$, and $\leq 1\%$, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

a certain amount of representative training data is available for model training. This work lead us to select three different deep learning-based architectures already proposed in the literature, *i.e.* ResNet-50 [32] (the same used in the evaluation of Section 5.2.1), Inception-Resnet V1 [77] and the recent Vision Transformer (ViT) [20].

The architecture choice is because of ResNet-50, as mentioned before, revealing high accuracy in several classification tasks, while Inception-Resnet V1 has been effectively used in [1] for the S-MAD task. Differently, the ViT model is an architecture recently proposed in the literature, that seems to be able to overcome the performance of traditional Convolutional Neural Networks (CNNs) for image classification [31]. We also internally tested other architectures obtaining lower results, here not reported for simplicity.

The experimental results reported in Table 5.4 show that the Inception-Resnet V1 outperforms the other architectures by a clear margin, with equal training and testing data, thus confirming the findings reported in [1]. Presumably, the presence of kernels with different sizes at the same level of the network enhances the ability of the model to detect specific patterns on pixel values, and then morphed images. Therefore, all the following experiments are performed using the Inception-Resnet model.

5.2.3 Investigation on data augmentation

Morphing Alg.	No augmentation				JPEG [83]				Print & Scan [27]			
	EER	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
UBO	.003	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**
OpenCV	**.008**	**.000**	**.000**	.006	.009	**.000**	**.000**	.005	**.008**	**.000**	**.000**	**.003**
FaceMorpher	.004	**.000**	**.000**	.003	**.004**	**.000**	**.000**	**.000**	.011	**.000**	**.000**	.012
StyleGAN	.013	**.000**	**.000**	.021	.011	**.000**	**.000**	.011	**.008**	**.000**	**.000**	**.008**
AMSL	.005	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	**.000**	.050	.050	.050	.300
Webmorph	.250	.350	.650	.900	**.158**	**.200**	**.250**	**.750**	.315	.500	.500	**.750**
Sqirlz Morph_D	.024	.012	.025	.123	**.001**	**.000**	**.000**	**.001**	.069	.019	.118	.192
FaceFusion	.114	.136	.243	.502	**.094**	**.088**	**.176**	**.408**	.129	.178	.341	.671
NTNU	.114	.132	.261	.519	**.091**	**.081**	**.176**	**.469**	.167	.313	.501	.787
UTW	**.312**	.641	.769	.942	.391	.740	.850	.964	.329	**.621**	**.741**	**.884**
FaceFusion_C	.125	.165	.351	.669	**.114**	**.134**	**.283**	**.633**	.162	.255	.434	.735
NTNU_C	.158	.265	.448	.703	**.149**	**.239**	**.404**	**.679**	.203	.412	.579	.829
UTW_C	.315	.643	.753	**.918**	**.298**	**.604**	**.736**	**.918**	.300	.622	.752	.940
WAED $\downarrow$	.3915				**.3515**				.4580			
Sqirlz $\text{Morph}_{P\&S}$	.252	.455	.540	.760	**.197**	**.320**	**.385**	**.520**	.218	.420	.505	.710
Sqirlz $\text{Morph}_{P\&S+JP2}$	.278	.495	.615	.855	.237	**.355**	.555	.780	**.210**	.395	**.475**	**.610**
$\textbf{WAED}_{P\&S}$ $\downarrow$	.5655				**.4530**				.4669			

Table 5.5: Morphing detection scores across different Data Augmentation techniques, given a fixed architecture (Inception-Resnet V1) and a Face Detector (MTCNN). Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq$ 10%, $\leq$ 5%, and $\leq$ 1%, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

Following the considerations reported in [27], we analyze the impact of different data augmentation techniques on the final classification accuracy.

Data augmentation techniques play a crucial role in many different classification tasks, increasing the amount and the variety of images available for model training. The context of face morphing is, in some respects, different from other applications since the morphing process leaves only labile traces and the risk of weakening such details by applying transformations to the original images is concrete.

Then, we evaluate here different techniques for data augmentation: some of them are the typical approaches used in the literature. In particular, we evaluate here image resizing, which is generally required for model training since the large majority of neural networks receive input images with a fixed spatial resolution; the tests are aimed at evaluating the impact of the resizing algorithm used (*i.e.* the interpolation algorithm) on the final accuracy of the model.

We also evaluate other transformations specific to this application scenario, defined taking into account the typical pipeline of the document issuing process. In many countries, in fact, the digital photo acquired by professional photographers

is printed on paper and then scanned to be included in the document during the eMRTD issuing process. Moreover, when stored into the chip, the image is compressed, either using the JPEG Sequential Baseline (ISO/IEC 10918-1) mode of operation or the JPEG-2000 Part-1 Code Stream Format (ISO/IEC 15444-1) [83]. Considering the minimum image size requirement of 11 kB given in [33], most of the issuing authorities adopt a compressed image size of around 12-15 kB; we follow here the approach adopted in [59] setting the maximum size of the compressed photo to 15 kB.

As to the printing and scanning process, we apply here the simulation approach introduced and described in [27].

The MAD results obtained using different data augmentation techniques are reported in Table 5.5. The first column represents the baseline, where no data augmentation is applied; the second column contains the results of a JPEG compression with a probability of 50% on input images; finally, the third column represents the performance obtained by applying the printing and scanning simulation with a probability of 50% on input images.

The results (and the corresponding WAED metric) are reported separately for the testing datasets used in the previous tables and for the P&S ones (not used in the previous experiments).

Mainly guided by the findings in [27], all augmented models were obtained by fine-tuning the baseline model, rather than training from scratch. As expected, data augmentation has a slight but noticeable effect on the performance of the model with respect to digital images. JPEG compression seems to produce in general a positive effect even on non-compressed and printed/scanned datasets.

The simulation of the P&S process is expected to produce positive effects on the P&S datasets and the results prove that in this case the model trained using the simulation of the printing and scanning process performs better than the model without this kind of augmentation; however, the advantages with respect to the model trained with JPG compression augmentation are quite limited. As to this aspect, we believe that the effectiveness of the simulation might be improved by an optimization of its parameters that should be tuned to better represent the real P&S process.

Finally, we internally test the investigated MAD also considering different color spaces in input, following the findings reported in [60] that highlight that the use of color spaces other than RGB might have a positive impact on MAD performance. Then, we convert all training and test images in grayscale, HLS and YCbCr color spaces: we omit to report the related Table since results reveal that the RGB representation offers the best performance, and indeed the color information positively contributes to the detection of morphed images. With grayscale images, we obtain WAED = 0.3824, with HSL WAED = 0.5677 and

with YCbCr WAED = 0.4360. We also tested a single channel in input, obtaining similar results (WAED = 0.4411 using only the L channel of HSL, WAED = 0.7453 using the Y channel of YCbCr color space).

5.2.4 Investigation on forensic features

Morphing Alg.	Fourier [21]				Wavelets [29]				PRNU [12]			
	EER	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
UBO	.112	.117	.235	.318	**.021**	**.000**	**.000**	**.063**	.125	.178	.244	.481
OpenCV	.083	.055	.156	.430	**.048**	**.022**	**.046**	**.186**	.066	.041	.083	.199
FaceMorpher	.052	.022	.056	.095	**.021**	**.004**	**.010**	**.042**	.077	.063	.106	.241
StyleGAN	.060	.037	.074	.269	.051	.025	.052	.170	**.023**	**.003**	**.013**	**.053**
AMSL	.403	.850	.850	.950	**.200**	**.250**	**.450**	**.600**	.356	.800	.850	.950
Webmorph	.500	.950	.950	1.000	**.411**	**.600**	**.650**	**.900**	.550	.900	.950	1.000
Sqirlz Morph_D	.163	.292	.402	.763	**.071**	**.070**	**.149**	**.356**	.260	.442	.550	.639
FaceFusion	.291	.613	.786	.922	**.262**	**.515**	**.680**	**.915**	.485	.925	.965	.989
NTNU	**.184**	.338	.507	.794	.191	**.327**	**.490**	**.761**	.246	.587	.813	.978
UTW	.879	1.000	1.000	1.000	.833	.998	1.000	1.000	**.159**	**.253**	**.410**	**.851**
FaceFusion_{JPG}	.446	.845	.922	.983	**.188**	**.325**	**.480**	**.727**	.372	.700	.832	.955
NTNU_{JPG}	.558	.927	.970	.993	**.239**	**.451**	**.585**	**.810**	.426	.835	.923	.986
UTW_{JPG}	.378	.757	.848	.976	.392	.774	.863	.963	**.368**	**.727**	**.846**	**.958**
WAED ↓	.7345				**.5768**				.7075			

Table 5.6: Morphing detection scores across different forensic features used in combination with the Inception-Resnet V1 architecture and the MTCNN Face Detection. Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq 10\%$, $\leq 5\%$, and $\leq 1\%$, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

The use of forensic features has received increasing attention not only in fake face image detection (the so-called *DeepFakes* [30]), but also in the MAD field [49]. Indeed, we implement in our framework a selection of the most used forensic features in the MAD task available in the literature.

As reported in [21], the *Fourier* transform can be effectively exploited to detect fake facial images; in [47] this feature is used to detect morphed images and then is implemented and tested in the Revelio framework.

Following the considerations reported in [68], the second investigation regards the use of the *Photo Response Non Uniformity* (PRNU) [12], *i.e.* the unique pattern noise related to a specific digital sensor used to acquire an image. The underlying idea is that the morphing procedure can affect the uniformity of the sensor noise, and then its analysis can help to spot morphed images.

Thirdly, our experiments aim to investigate the use of *wavelets* [29], since in [1] an approach based on an attention-aware neural network that receives in input this

kind of feature is presented, obtaining an interesting accuracy on the NIST FRVT MORPH [48] platform. We implement this approach to the best of our knowledge[3], following two different approaches.

In both experiments, following the paper [11], we apply three-level undecimated 2D wavelet decomposition, using *Daubechies* 4 (db4) as the mother wavelet; in the first implementation, a one-level wavelet decomposition is applied, while in the second we apply a three-level decomposition and we finally exploit a selection of 23 sub-bands channel-wise stacked. The first approach provided better results in our experiments, so the metrics are reported only for this implementation.

Experimental results are reported in Table 5.6: on our testing set all the forensic features seem to have only a limited capability in detecting morphed faces produced by morphing algorithms never seen during the training procedure. Specifically, results suggest a limited generalization capability in the cross-morphing algorithm scenario, with a lower EER on the first set of testing datasets (in which the same morphing algorithm is also used in the training procedure), with respect to the EER achieved in the second, more challenging, block of testing datasets. Best performances across different forensic features are provided by the use of wavelets with one-level decomposition (WAED = 0.577, while the three-level decomposition achieves WAED = 0.603). This finding has a confirmation in [1], in which an Inception-Resnet V1 architecture achieves comparable performance receiving in input RGB images or wavelets.

5.2.5 Investigation on training data

Here, we investigate the influence of training data, and in particular the availability of different morphing algorithms, in the development of robust MAD methods.

We train the best MAD detector obtained, *i.e.* the Inception-Resnet V1 network receiving for input RGB faces detected with MTCNN, on different training data configurations. This experimental validation is useful in order to understand how the image visual quality, the variety of morphing algorithms, and the amount of training data influence the final performance of the system.

As expected, the results reported in Table 5.7 reveal that the combination of all available datasets produces the best performance, thus highlighting the importance to train MAD models on varied and large-size datasets.

In particular, the presence of different morphing algorithms is a key element, even when they generate images with visible artefacts and, from a general point of view, produce low-quality morphed images (*e.g.* morphed faces produced with WebMorph, FaceMorpher and OpenCV morphing algorithms). As to this point, we

[3]The original paper is currently patent pending, and then a limited amount of implementation details are revealed.

Morphing Alg.	UBO [6]			OpenCV [44]			FaceMorpher [55]			StyleGAN [36]		
	EER	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**EER**	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
UBO	**.000**	**.000**	**.000**	.071	.079	.253	.152	.298	.465	.211	.537	.747
OpenCV	.046	.043	.334	**.022**	**.006**	**.041**	.060	.069	.155	.135	.421	.702
FaceMorpher	.024	**.006**	.159	**.020**	**.006**	**.027**	.035	.030	.080	.168	.509	.849
StyleGAN	.171	.833	.987	.057	.082	.281	.124	.254	.535	**.004**	**.000**	**.003**
AMSL	**.022**	**.000**	.250	.100	.100	.200	.050	.050	**.150**	.250	.500	.700
Webmorph	.405	.900	1.000	**.300**	.700	.850	**.300**	**.550**	**.750**	.500	.900	.950
Sqirlz Morph_D	**.020**	**.011**	.151	.044	.048	**.146**	.252	.413	.589	.150	.238	.460
FaceFusion	.407	.925	.990	**.176**	**.442**	**.680**	.181	.445	.727	.321	.798	.955
NTNU	.307	.889	.990	.137	.330	.611	**.114**	**.239**	**.510**	.310	.776	.945
UTW	.312	.800	.943	**.204**	**.645**	**.904**	.363	.859	.969	.403	.854	.955
FaceFusion_{JPG}	.174	.552	.834	**.141**	**.342**	**.608**	.146	.357	.659	.316	.771	.921
NTNU_{JPG}	.215	.688	.889	.183	.478	**.691**	**.174**	**.450**	.745	.353	.816	.952
UTW_{JPG}	.405	.898	.981	**.344**	**.776**	**.935**	.365	.844	.965	.456	.925	.987
WAED $\downarrow$	.5815			**.4271**			.4675			.6672		

Table 5.7: Morphing detection scores across different training sets given a fixed model (Inception-Resnet V1) and face detector (MTCNN). Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq 5\%$ and $\leq 1\%$, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1). Due to space reasons, neither the value of the lowest BPCER related to APCER $\leq 10\%$, nor the column containing the results of training with the combined datasets (see Table 5.4) is reported.

have to consider that the face region is cropped after detection and most of such artefacts are cut off; this allows us to exploit for training the features of the facial region without relying on the heavy presence of artefacts in the region surrounding face (which is unlikely in a real operational scenario).

Moreover, results reported in the top part of Table 5.7, confirm the tendency of MAD approaches to overfit on the training dataset, as also reported in [59, 58]. Indeed, in all cases, best performances are obtained when the morphing algorithms used in training and testing correspond, with the exception of OpenCV and FaceMorpher algorithms, which produce similar morphed images.

These considerations highlight the importance of cross-dataset evaluations in the MAD field, in combination with results obtained on sequestered datasets [19, 48].

5.2.6 Test on FVC-onGoing platform

Finally, we test the developed S-MAD methods on the SOTAMD sequestered datasets [59] through the FVC-onGoing [19] platform.

In particular, following the experimental results, we test different versions of a solution based on the Inception-Resnet V1 pre-trained on the ImageNet dataset,

	Method	R-1	R-2	R-3	R-2$_{PS}$
Training	**UBO**	✓	✓	✓	✓
	OpenCV	✓	✓	✓	✓
	FaceMorpher	✓	✓	✓	✓
	StyleGAN	✓	✓	✓	✓
	AMSL		✓	✓	✓
	WebMorph		✓	✓	✓
	Sqirlz		✓	✓	✓
	FaceFusion		✓	✓	✓
	NTNU		✓	✓	✓
	UTW		✓	✓	✓
Augm.	**JPEG**			✓	
	P&S				✓

Table 5.8: Training configuration for the different versions of our method tested on the FVC-onGoing [5] platform.

which receives input faces cropped with the MTCNN face detector.

The first version (referred to as "R-1") is trained on the morphing algorithms exploited to create the training set of all previous experiments, *i.e.* UBO, OpenCV, FaceMorpher, and StyleGAN, following the 80-20 split for the training and validation procedure. The second version ("R-2") is trained on all data available in the Revelio framework, thus including, in addition to the previous ones, the morphed images obtained with AMSL, WebMorph, Sqirlz Morph, FaceFusion, NTNU, and UTW algorithms. Since the amount of data is increased, we split train and validation sets with 90% and 10% percentages. The third version of the method ("R-3") is the same as the previous one (R-2), and the JPEG compression (see Section 5.2.3) is randomly applied to input data during the training phase. Finally, the last version ("R-2$_{PS}$") is the same as R-2 but trained by applying the P&S simulation process [27] on input images. Only in this case, the model starts the training with parameters that belong to R-2. A summary of these settings is reported in Table 5.8.

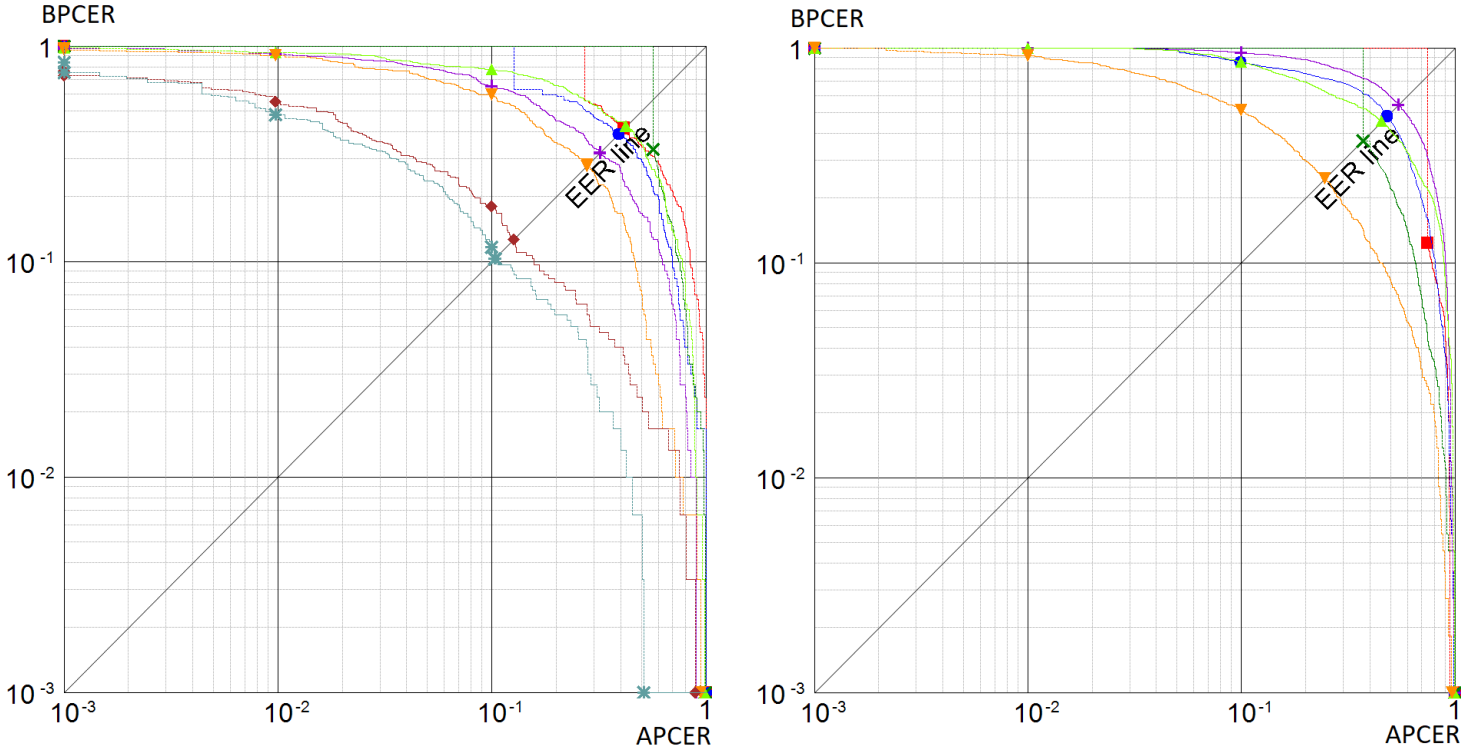
Results are shown in Table 5.9 and also officially published on the platform. It is worth noting that R-1 achieves state-of-the-art results, despite the limited amount and variety of training data that belong to publicly released datasets. R-2 confirms the tendency to have better performance when new, and possibly high-quality, morphed images are available during the training procedure, probably due to also the presence of similar morphing algorithms in the test set [59]. The efficacy of the JPG compression, as observed in the Revelio experimental evaluation,

Algorithm	Year	EER	$B_{0.1}$	$B_{0.05}$	$B_{0.01}$
[38]	2017	42.32	78.00	82.67	93.33
[76]	2018	41.38	100	100	100
[60]	2020	31.80	65.00	79.33	91.67
[27]	2021	38.99	100	100	100
R-1	2023	27.77	59.33	70.67	90.33
R-2	2023	12.67	18.00	28.33	55.00
R-3	2023	**10.33**	**11.67**	**23.67**	**48.00**
[57]	2017	54.37	94.89	98.27	99.91
[38]	2017	45.52	85.86	96.90	100
[76]	2018	43.34	100	100	100
[68]	2019	48.04	85.86	97.35	100
[27]	2021	37.10	100	100	100
R-2$_{\mathbf{PS}}$	2023	**24.63**	**51.28**	**68.25**	**91.42**

Table 5.9: Comparison of the results on the sequestered SMAD-SOTAMD_D-1.0 (top) and SMAD-SOTAMD_P&S-1.0 (bottom) benchmarks, respectively, through the FVC-onGoing [5] platform. As shown, S-MAD algorithms developed with *Revelio* framework outperform the competitors.

is confirmed by the results of R-3, proving the efficacy of the proposed framework to be an effective and valuable tool in the development and deployment of MAD algorithms. Similar observations are true also for the P&S morphed images: the P&S simulation algorithms implemented in the framework can be effectively used to create competitive solutions avoiding the time-consuming process of printing and scanning real photos.

The Detection Error Tradeoff (DET) curves computed on the SOTAMD sequestered dataset on the FVC-onGoing [5] platform are reported in Figure 5.1a and Figure 5.1b, with which is possible to appreciate the detail of the performance of the proposed systems and the competitors tested on digital (left) and P&S (right) morphed images. To summarize, overall results suggest that it is possible to use the Revelio framework to develop, in a simple and effective manner, state-of-the-art S-MAD systems, clearly improving the performance obtained by the competitors, also exploiting only publicly released datasets.



(a) SMAD-SOTAMD_D-1.0: competitor reported: R-1 (orange), R-2 (claret), R-3 (dark green), [27] (blue), [38] (light green), [76] (red), [60] (purple).

(b) SMAD-SOTAMD_P&S-1.0: competitor reported: R-2_{PS} (orange), [27] (dark green), [38] (light green), [76] (red), [60] (purple).

Figure 5.1: DET curves computed on the SOTAMD sequestered dataset on the FVC-onGoing [5] platform.

Chapter 6

Differential MAD experiments

In this Chapter, we define the training and testing protocols used in our experimental evaluation for D-MAD systems, describing the datasets used and the training and testing split. In particular, we aim to investigate the performance of composing both an S-MAD algorithm with a state-of-the-art D-MAD approach in the literature.

6.1 Datasets and protocols

Morphing Algorithm	**Train/Val (%)**	**Test (%)**
UBO [6]	90+10	0
OpenCV [44]	90+10	0
FaceMorpher [55]	90+10	0
StyleGAN [36]	90+10	0
Sqirlz Morph$_D$ [84]	90+10	0
Sqirlz Morph$_{P\&S}$ [84]	90+10	0
FaceFusion [22]	0	100
UTW [59]	0	100
NTNU [59]	0	100

Table 6.1: Morphing algorithms and datasets in the *Revelio* framework used for the experimental evaluation. For each morphing algorithm, the percentage of images available during the training, validation, and testing phases are shown.

Similarly to what is described in Section 5.1, we group the train and test datasets relying on the morphing algorithm used to produce morphed images, as also reported in the first column of Table 3.1.

All the experiments are carried out following the dataset organization and training/testing protocols described in Table 6.1. Table 3.3 reports, for each public face dataset, the number of bona fide and morphed couples. When training and evaluating the D-MAD experiments, the images from the CFD data source are not included, as very few subjects have an alternative pose that can be used as a live-capture image; therefore, no couples are available for that dataset. Therefore, the FaceFusion, UTW, and NTNU morphing algorithms are comprised exclusively of images from the FEI dataset (24000 morphed couples). Moreover, while bona fide couples are reported more than once in the aforementioned Table, no duplicate couples are present in either the training, validation, or test sets: this is done to ensure that the bona fide couples are present only once after merging the different data sources, thus preventing the bona fide class to have a greater weight during training.

Chapter 5 placed a greater focus on evaluating the performance of the system with respect to the employed morphing algorithm. However, the main objective of the following experiments is to assess the efficacy of each tested algorithm according to the identity in the live-capture image. More specifically, results are split into two distinct groups, without discriminating according to the employed morphing algorithm:

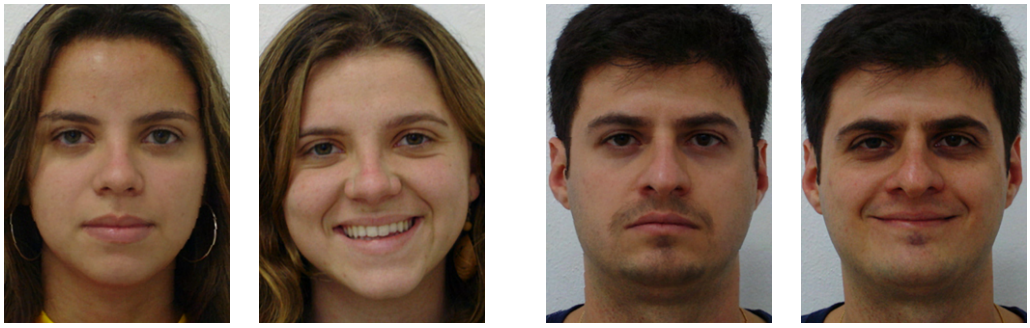
- *Criminal*: contains bona fide (*i.e.* no morphing attempt is present) and morphed couples; in the latter, the live image contains the criminal's identity;

- *Accomplice*: contains bona fide and morphed couples in which the live image contains the accomplice's identity; due to the greater similarity between the subjects present in both pictures, this group is generally considered more challenging than the previously mentioned one.

A sample for each kind of couple can be found in Figure 6.1. In any case, it is important to note that the subjects depicted in the live images do not have a neutral pose, to simulate how a real passenger would realistically behave when at the Automated Border Control (ABC) gate; therefore, the model used to extract the identity feature vector must be resistant variations in both pose and possibly lighting conditions as well.

Unless otherwise specified, training is performed on all available couples, *i.e.* bona fide and morphed, with both criminal and accomplice. While training, only images with a morphing factor of 0.5 are used.

Moreover, since the effectiveness of SVM classifiers for tackling the D-MAD task has been proven in the literature (see Section 4.2), we run each experiment twice, swapping the underlying classifier: indeed, we test both an MLP with varying architecture according to the employed features, and an SVM with RBF kernel.



(a) Criminal (b) Accomplice

Figure 6.1: Visual samples of two couples; note how the live picture (on the right) is not used to create the morphed image (on the left); moreover, the subject is depicted with a non-neutral pose, to simulate how a passenger would realistically behave at the gate.

In order to evaluate and compare the investigated D-MAD methods, we use the metrics reported in Sections 3.5 and 3.6.

Moreover, we employ the WAED metric (introduced in Section 3.6.1) to summarize and simplify the comparison of the diverse approaches. However, as only one test set is employed, we adopt $w_D = 1$. Finally, as we expect the two above-mentioned testing groups to yield substantially different results and to gather better insights into the performance of the tested models, we compute two distinct values for the WAED metric.

6.2 Experimental results

Following the established protocol in the previous Section, results are reported divided by type of subject present in the live image and by employed classifier.

To have a baseline against which to compare the efficacy of other D-MAD methods, in the first experiment (referred to as *S-MAD*), the R-3 model (illustrated in Section 5.2.6) is used as a feature extractor on the suspected morphed image, and the resulting vector is then used as the classifier's input, thus tackling the D-MAD problem as an S-MAD task.

Inspired by the work in [66], the second experiment (referred to as *ArcFace*) employs features that are extracted from both images using the *DeepFace* [72] implementation of the *ArcFace* [18] network. The embedding of the live image is subtracted from the embedding of the suspected morphed image, and the resulting vector is then used as the classifier's input.

Finally, to gather a better insight into the possible contribution of a state-of-the-art S-MAD algorithm on a D-MAD system, we run a third experiment (referred to as *Both*), in which we concatenate the features obtained from the two previous

methods and use the resulting vector as the classifier's input. Particular attention is devoted to the results of couples where the accomplice is present in the live image, which represents a more challenging setup due to the greater similarity between the two subjects.

Each SVM has been trained using an RBF kernel with a $C = 3$ regularization factor and a γ kernel coefficient which is inversely proportional to the variance of the training data received in input. Each MLP model is trained using the binary cross-entropy loss function. The chosen optimizer is *Adam*, with a learning rate equal to $5 \cdot 10^{-4}$ and early stopping (with patience of 5 epochs and a minimum improvement of 10^{-3}).

The experimental results obtained from running the three methods are reported in Table 6.2.

Feature	**Classifier**	**Fusion**	**Criminal**					**Accomplice**				
			WAED	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**WAED**	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
S-MAD	MLP	N/A	.359	.186	.255	.360	.515	.359	.186	.255	.360	.515
	SVM		.302	.175	.195	.250	.450	.302	.175	.195	.250	**.450**
ArcFace	MLP	N/A	.241	.085	.072	.147	.447	.509	.180	.300	.470	.827
	SVM		**.165**	**.066**	**.043**	**.085**	**.310**	.492	.175	.320	.475	.780
Both	MLP	Concat.	.350	.168	.225	.345	.520	.340	.168	.222	.317	.510
	SVM		.313	.175	.213	.275	.460	.305	.175	.198	.248	.458
	MLP	MM	.297	.138	.180	.265	.463	**.289**	**.132**	**.150**	**.245**	.463
	SVM		.325	.160	.233	.320	.475	.305	.140	.180	.278	.475
	MLP	MV	.389	.185	.292	.398	.563	.362	.175	.245	.338	.543
	SVM		.410	.257	.360	.423	.530	.344	.195	.245	.317	.495

Table 6.2: Morphing detection scores obtained on the FEI test set across different features, classifiers and feature fusion techniques. Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq 10\%$, $\leq 5\%$, and $\leq 1\%$, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

In particular, the first observation is that MLPs generally perform worse than SVMs when the features are standalone. It is also possible to notice that the metrics obtained on the S-MAD approach are identical regardless of the type of couple: this is to be expected, as in both cases the same suspected morphed images are used and therefore the same results are reported.

As anticipated, ArcFace provides considerably better performance when compared to the S-MAD method, indicating that a trusted, live-capture image proves to be effective in tackling the task by comparing the two extracted identities; however, the performance gap is significantly reduced when the accomplice is present in the live-capture image: this is probably due to the greater similarity between the identities, thus making it more challenging for the classifier to find an effective

class boundary.

Finally, the third method demonstrates that incorporating S-MAD features into a D-MAD algorithm provides a noticeable performance boost when the suspected morphed image is compared against the accomplice; nevertheless, the same performance improvement cannot be found when the passport picture is compared against the criminal. Moreover, the technique employed to merge the ArcFace and S-MAD features proves to be crucial in order to obtain satisfactory performance. In particular, a naive concatenation of the two features provides results that are remarkably similar to those reported for the S-MAD-only approach. This behavior may suggest that the contribution of S-MAD features in correctly classifying the couples is remarkably strong, outweighing the features provided by ArcFace and thus negating the benefits they bring in comparing the identities.

To further investigate this behavior, we run a *t-distributed Stochastic Neighbor Embedding* (t-SNE) [80] dimensionality reduction on the input features, divided both by source (*i.e.* ArcFace or S-MAD) and by ground truth (*i.e.* bona fide or morphed). The resulting plot, shown in Figure 6.2, highlights how the features can easily be separated by their respective source, suggesting that they may occupy different portions of the feature space. However, there is no clear separation between bona fide and morphed feature vectors; this reinforces the hypothesis that the classifier could be prioritizing the S-MAD features while disregarding those generated by ArcFace.

In order to try to overcome these issues, we test two different fusion strategies:

- *Min-max* (MM): before concatenating the two feature vectors, they are separately rescaled to have each component in the $[0, 1]$ range;
- *Mean-variance* (MV): before concatenating the two feature vectors, they are separately rescaled to have each component with mean value $\mu = 0$ and variance $\sigma = 1$.

Firstly, the performance gap that was previously found between MLPs and SVMs is not present when the two features are merged together; indeed, the former almost always outperforms the latter.

Secondly, the MV strategy provides unsatisfactory results, which are worse than the naive concatenation strategy. An ex-post numerical analysis on the normalized feature vectors used for training shows that, even when each component is rescaled to have $\mu = 0$ and $\sigma = 1$, the ArcFace and S-MAD vectors still show significant differences in range. This could be a possible explanation of the performance of the MM fusion strategy, thus proving that translating the two feature vectors to the same numeric range helps improve the model's performance.

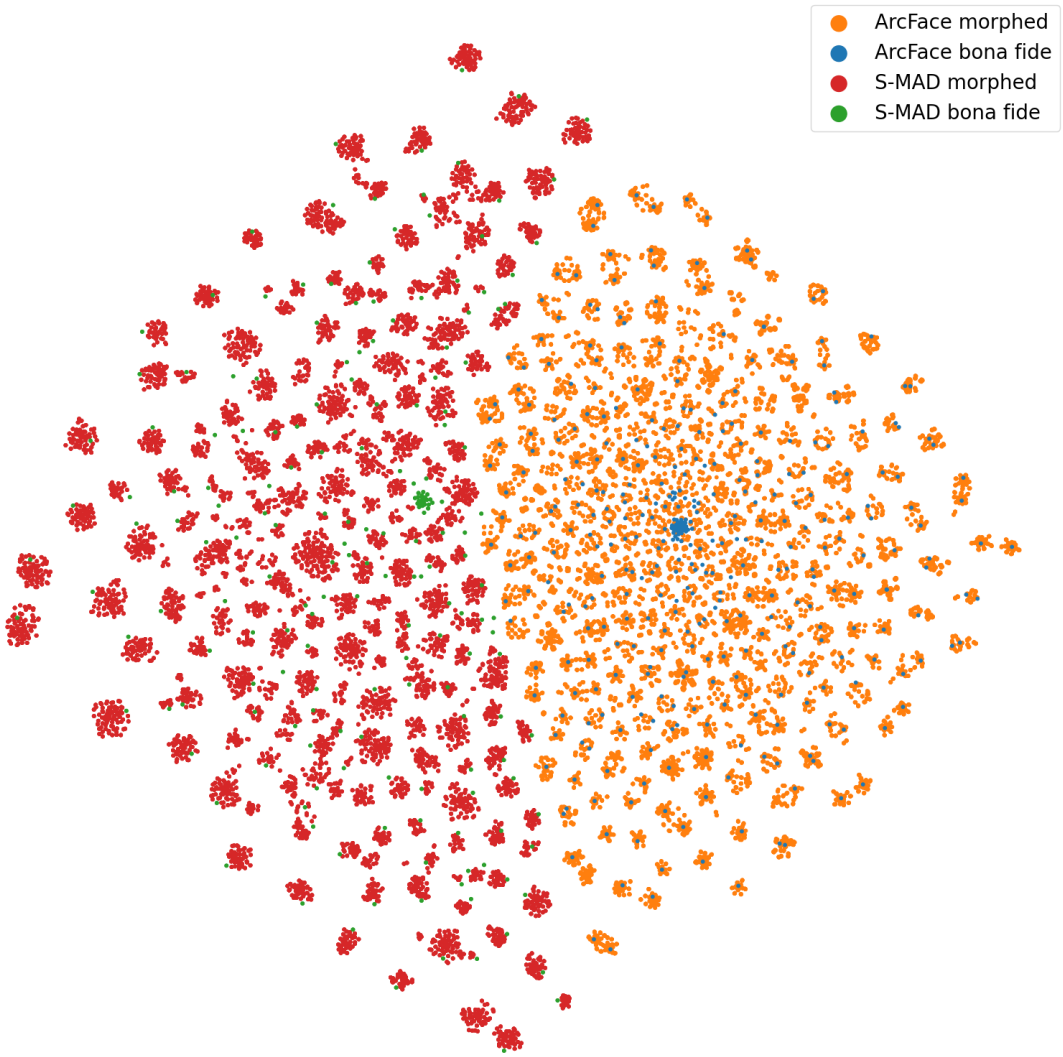


Figure 6.2: *t-distributed Stochastic Neighbor Embedding* (t-SNE) [80] of ArcFace and S-MAD feature vectors divided by ground truth. The blue and orange dots respectively represent the bona fide and morphed ArcFace embeddings, while the green and red dots respectively represent the bona fide and morphed S-MAD embeddings. The different feature vectors can easily be separated by source (*i.e.* ArcFace versus S-MAD), but not by ground truth (*i.e.* bona fide versus morphed).

6.2.1 Investigation on cosine distance

Next, we test the performance impact of including the cosine distance between the two ArcFace feature vectors. The underlying idea comes from the fact that, as illustrated in [18] and in Figure 4.7, the embeddings produced by the model are optimized in such a way that the geodesic angle between each identity is maximized. Therefore, the cosine distance between the embeddings obtained from both the suspected morphed and live images should be small (*i.e.* approximately 1) when no morphing algorithm is applied; on the contrary, if the distance is greater (*i.e.* closer to -1), then we can assume that the two presented identities are too far apart and therefore some morphing process has taken place.

To determine if there is a tangible performance improvement, we train an MLP whose input is composed of both the ArcFace and S-MAD features with *min-max* fusion strategy, as well as the cosine distance between the two ArcFace embeddings. Moreover, because of the chosen fusion strategy, we investigate whether to translate the cosine distance from its $[-1, 1]$ range to $[0, 1]$. Experimental results are shown in Table 6.3.

Cosine dist.	**Criminal**					**Accomplice**				
	WAED	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$	**WAED**	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
None	.297	.138	.180	.265	.463	.289	.132	**.150**	.245	**.463**
$[-1, 1]$	**.275**	**.125**	**.147**	**.235**	**.440**	**.288**	**.125**	.155	**.237**	.468
$[0, 1]$	.307	.141	.188	.290	.470	.290	.132	.170	**.237**	.465

Table 6.3: Morphing detection scores obtained on the FEI test set with and without employing the cosine distance. Results are reported in terms of Equal Error Rate (EER), the lowest BPCER related to APCER $\leq 10\%$, $\leq 5\%$, and $\leq 1\%$, respectively. The proposed WAED metric summarizes performance (lower is better) across listed testing datasets (see Section 3.6.1).

Experimental results show that adding the cosine distance provides a tangible performance improvement only when left in its original range. On the contrary, if the cosine distance is translated into the $[0, 1]$ range the model's performance is considerably worsened.

6.2.2 Test on the FVC-onGoing platform

Finally, we test the developed D-MAD methods against the SOTAMD sequestered test set through the FVC-onGoing [5] platform.

In particular, we choose three different baselines and then test two different solutions. The first baseline is the algorithm depicted in [66] and summarized in Section 4.2.3: this method is chosen for being the state of the art on the SOTAMD

Algorithm	**Year**	**EER**	$\mathbf{B}_{0.1}$	$\mathbf{B}_{0.05}$	$\mathbf{B}_{0.01}$
[25]	2018	14.17	17.20	22.77	64.57
[66]	2020	**4.54**	**2.00**	**3.93**	**18.87**
S-MAD	2023	9.66	9.67	22.67	46.00
RD-1	2023	10.40	10.93	25.60	48.00
RD-2	2023	10.23	10.33	19.67	47.47

Table 6.4: Comparison of the results on the sequestered DMAD-SOTAMD_D-1.0 benchmark, through the FVC-onGoing [5] platform.

sequestered test set and to gather better insight on the influence of a state-of-the-art S-MAD model when paired with the ArcFace features. The second baseline is the method described in [25] and outlined in Section 4.2.2: this way, we are able to compare the performance of our methods with a different approach that is not based on machine learning but rather on *face demorphing*, *i.e.* the inversion of the morphing process. Finally, the third chosen baseline is represented by running the R-3 model obtained in Section 5.2.6 only on the suspected morphed image, thus tackling the D-MAD task as an S-MAD problem.

The first tested algorithm (referred to as "RD-1") is obtained by naively concatenating the unnormalized ArcFace and S-MAD features. Moreover, after the results shown in Table 6.3, we add the cosine distance of the two ArcFace vectors as an extra feature. The second version ("RD-2") is a variation of RD-1, where both S-MAD and ArcFace features are normalized following the *min-max* strategy depicted in Section 6.2 (*i.e.* both feature vectors are separately rescaled to the $[0, 1]$ range) while leaving the cosine distance in its natural range (*i.e.* $[-1, 1]$).

Results are shown in Table 6.4. As reported, the state-of-the-art method proposed in [66] provides the best overall performance, while only focusing on the identities of the two presented subjects and disregarding any possible artifact that might be present in the suspected morphed image. However, one noteworthy aspect of these results is that both the proposed methods (RD-1 and RD-2) fail to outperform the S-MAD approach, thus suggesting that while the employed features do contain some information regarding the identities and the morphing process, these are probably combined in a suboptimal way; therefore, further work must be done to investigate on this issue and to find a better feature fusion method that contains more predictive power. Moreover, the RD-2 approach provides a slight improvement in all metrics when compared to the original version (RD-1), thus showing that the applied feature scaling described in Section 6.2 does bring a slight performance improvement. Finally, another interesting aspect is that both methods outperform the algorithm proposed in [25], showing that machine learning-based techniques yield overall better results.

The Detection Error Tradeoff (DET) curve computed on the SOTAMD sequestered dataset on the FVC-onGoing [5] platform is reported in Figure 6.3, with which is possible to appreciate the detail of the performance of the proposed systems and the competitors tested on digital images.

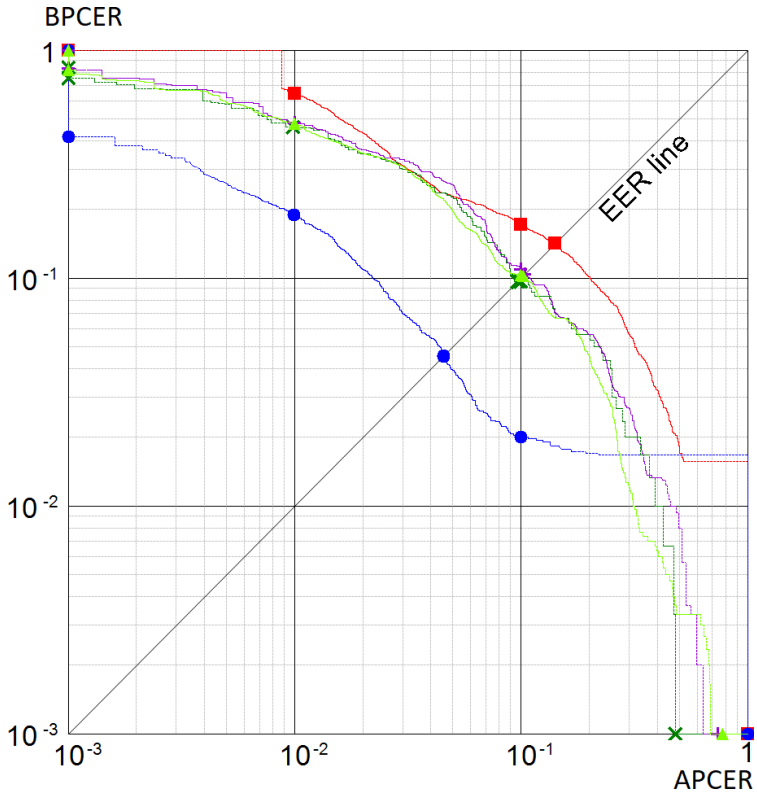


Figure 6.3: DET curve computed on the DMAD-SOTAMD_D-1.0 benchmark on the FVC-onGoing [5] platform. Competitor reported: RD-1 (claret), RD-2 (light green), S-MAD (dark green), [25] (red), [66] (blue).

Chapter 7

Conclusions

This thesis presents Revelio, a new framework aimed at providing effective support for the development, training, and evaluation of MAD algorithms. The framework is publicly available on GitHub[1] and released under the Apache-2.0 license. Our extensive experimentation confirms that Revelio allows the user to develop and test MAD approaches in a simple and effective way, achieving state-of-the-art results on sequestered datasets.

Several considerations can be expressed after the analysis of the experimental evaluation.

Firstly, both S-MAD and D-MAD are confirmed to be challenging tasks, and the accuracy of existing MAD methods still does not satisfy real-world operational requirements. The lack of a probe image with which to compare the tested image is a key element for the final performance; this is confirmed in the literature and our experimental validation, where state-of-the-art D-MAD methods usually achieve greater accuracy in detecting morphed images than S-MAD algorithms.

Secondly, experimental results suggest that the availability of a great amount and variety of training data, including several morphing algorithms and subjects belonging to different source datasets, is an important element to improve S-MAD performance. We believe that, in this context, the understanding of newly proposed MAD systems might be significantly improved by the possibility of sharing a common set of training datasets in combination with tests on public datasets and, in particular, on sequestered datasets hosted in public platforms [48, 5]. The adoption of the Revelio framework and the WAED metric can reduce the effort needed to develop new MAD systems and to test and compare them with other related approaches.

Another point of attention, still difficult to address, is the printing and scanning process which makes the problem much more challenging, especially for S-MAD, in

[1] `https://github.com/ndido98/revelio`

particular when followed by a compression step often applied to meet the image size limits in eMRTD chips.

Finally, while state-of-the-art D-MAD algorithms do provide better performance than their S-MAD counterpart, they tend to focus almost exclusively on the presented identities, while disregarding any artifacts that the morphing process may leave on the passport image.

A great variety of future work can be planned: firstly, continuous maintenance and documentation activities related to the Revelio framework will be done in order to support the whole community of the iMARS project, funded by the European Union's Horizon 2020 research and innovation program; secondly, new state-of-the-art MAD approaches will be implemented, so that all research groups will be able to compare their results against the current literature; thirdly, a more in-depth investigation into the use of forensic features in the S-MAD task will be done, with particular attention to the use of undecimated wavelet decompositions, which obtained remarkable results in the NIST FRVT MORPH [48] and, on that regard, the framework will be updated to more easily support submitting a proposed algorithm to NIST; moreover, after the above-mentioned upgrades, the best model we obtained in our experimental evaluations will be sent to NIST for benchmarking; finally, to overcome the limitation of D-MAD systems that do not take into account the presence of morphing artifacts, new ways of combining features must be investigated so that the resulting predictive power is greater than the two separate ones. This thesis, developed thanks to the European Union's Horizon 2020 research and innovation program, resulted in a scientific publication (parts of which are reported verbatim in this thesis) that is currently under submission for the *Expert Systems With Applications* (ESWA)[2] journal, and we will integrate our work with the constructive comments that will arise from the revisions of our submission.

[2] `https://www.sciencedirect.com/journal/expert-systems-with-applications`

Acknowledgements

This work is part of the iMARS project. The project received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 883356. Disclaimer: 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.

Bibliography

[1] Poorya Aghdaie et al. 'Attention aware wavelet-based detection of morphed face images'. In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2021, pp. 1–8.

[2] Poorya Aghdaie et al. 'Detection of morphed face images using discriminative wavelet sub-bands'. In: *2021 IEEE International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2021, pp. 1–6.

[3] Brandon Amos, Bartosz Ludwiczuk and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. CMU-CS-16-118, CMU School of Computer Science, 2016.

[4] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. 'SURF: Speeded Up Robust Features'. In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof and Axel Pinz. Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8. DOI: 10.1007/11744023_32.

[5] Biolab. *FVC-onGoing*. URL: https://biolab.csr.unibo.it/fvcongoing/ (visited on 30/11/2022).

[6] Biolab. *Morphed Face Generation Tools*. URL: https://biolab.csr.unibo.it/morphedfacegenerationtools.html (visited on 30/11/2022).

[7] Guido Borghi et al. 'A double Siamese framework for differential morphing attack detection'. In: *Sensors* 21.10 (2021), p. 3466.

[8] Guido Borghi et al. 'Incremental Training of Face Morphing Detectors'. In: *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society. 2022, pp. 914–921.

[9] Qiong Cao et al. 'VGGFace2: A dataset for recognising faces across pose and age'. In: *International Conference on Automatic Face and Gesture Recognition*. 2018.

[10] Yin-Wen Chang et al. 'Training and Testing Low-degree Polynomial Data Mappings via Linear SVM'. In: *Journal of Machine Learning Research* 11.48 (2010), pp. 1471–1490.

[11] Baaria Chaudhary et al. 'Differential morph face detection using discriminative wavelet sub-bands'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1425–1434.

[12] Mo Chen et al. 'Source digital camcorder identification using sensor photo response non-uniformity'. In: *Security, steganography, and watermarking of multimedia contents IX*. Vol. 6505. SPIE. 2007, pp. 517–528.

[13] L P Chew. 'Constrained Delaunay Triangulations'. In: *Proceedings of the Third Annual Symposium on Computational Geometry*. SCG '87. Association for Computing Machinery, 1987, pp. 215–222. ISBN: 0897912314. DOI: 10.1145/41958.41981.

[14] Corinna Cortes and Vladimir Vapnik. 'Support-vector networks'. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. DOI: 10.1007/BF00994018.

[15] Luca Debiasi et al. 'PRNU-based detection of morphed face images'. In: *2018 International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2018, pp. 1–7.

[16] Lisa DeBruine and Benedict Jones. 'Face research lab London set'. In: *Psychol. Methodol. Des. Anal* (2017).

[17] Jia Deng et al. 'Imagenet: A large-scale hierarchical image database'. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[18] Jiankang Deng et al. 'ArcFace: Additive Angular Margin Loss for Deep Face Recognition'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4685–4694. DOI: 10.1109/CVPR.2019.00482.

[19] Bernadette Dorizzi et al. 'Fingerprint and on-line signature verification competitions at ICB 2009'. In: *Advances in Biometrics* (2009). Ed. by Massimo Tistarelli and Mark S Nixon, pp. 725–732. DOI: 10.1007/978-3-642-01793-3_74.

[20] Alexey Dosovitskiy et al. 'An image is worth 16x16 words: Transformers for image recognition at scale'. In: *arXiv preprint arXiv:2010.11929* (2020).

[21] Ricard Durall et al. 'Unmasking DeepFakes with simple features'. In: *arXiv preprint arXiv:1911.00686* (2019).

[22] FaceFusion. *FaceFusion*. URL: http://www.wearemoment.com/FaceFusion/ (visited on 30/11/2022).

[23] Matteo Ferrara and Annalisa Franco. 'Morph Creation and Vulnerability of Face Recognition Systems to Morphing'. In: *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*. Ed. by Christian Rathgeb et al. Springer International Publishing, 2022, pp. 117–137. ISBN: 978-3-030-87664-7. DOI: 10.1007/978-3-030-87664-7_6. URL: https://doi.org/10.1007/978-3-030-87664-7_6.

[24] Matteo Ferrara, Annalisa Franco and Davide Maltoni. 'Decoupling texture blending and shape warping in face morphing'. In: *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2019, pp. 1–5.

[25] Matteo Ferrara, Annalisa Franco and Davide Maltoni. 'Face Demorphing'. In: *IEEE Transactions on Information Forensics and Security* 13.4 (2018), pp. 1008–1017. DOI: 10.1109/TIFS.2017.2777340.

[26] Matteo Ferrara, Annalisa Franco and Davide Maltoni. 'Face demorphing'. In: *IEEE Transactions on Information Forensics and Security* 13.4 (2017), pp. 1008–1017.

[27] Matteo Ferrara, Annalisa Franco and Davide Maltoni. 'Face morphing detection in the presence of printing/scanning and heterogeneous image sources'. In: *IET Biometrics* 10.3 (2021), pp. 290–303.

[28] Matteo Ferrara, Annalisa Franco and Davide 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. DOI: 10.1109/BTAS.2014.6996240.

[29] Amara Graps. 'An introduction to wavelets'. In: *IEEE computational science and engineering* 2.2 (1995), pp. 50–61.

[30] David Güera and Edward J Delp. 'DeepFake video detection using recurrent neural networks'. In: *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE. 2018, pp. 1–6.

[31] Kai Han et al. 'A survey on visual transformer'. In: *arXiv preprint arXiv: 2012.12556* 2.4 (2020).

[32] Kaiming He et al. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[33] *Information technology — Extensible biometric data interchange formats — Part 5: Face image data*. Standard. International Organization for Standardization, 2019.

[34] Anil Jain, Brendan Klare and Arun Ross. 'Guidelines for best practices in biometrics research'. In: *2015 International Conference on Biometrics (ICB)*. 2015, pp. 541–545. DOI: 10.1109/ICB.2015.7139116.

[35] Juho Kannala and Esa Rahtu. 'BSIF: Binarized statistical image features'. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 1363–1366.

[36] Tero Karras et al. '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.

[37] Davis E King. 'Dlib-ml: A Machine Learning Toolkit'. In: *Journal of Machine Learning Research* 10 (2009), pp. 1755–1758.

[38] Christian Kraetzer et al. 'Modeling attacks on photo-ID documents and applying media forensics for the detection of facial morphing'. In: *Proceedings of the 5th ACM workshop on information hiding and multimedia security*. 2017, pp. 21–32.

[39] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. 'Imagenet classification with deep convolutional neural networks'. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[40] Shengcai Liao et al. 'Learning Multi-scale Block Local Binary Patterns for Face Recognition'. In: *Advances in Biometrics*. Ed. by Seong-Whan Lee and Stan Z. Li. Springer Berlin Heidelberg, 2007, pp. 828–837. ISBN: 978-3-540-74549-5. DOI: 10.1007/978-3-540-74549-5_87.

[41] Ziwei Liu et al. 'Deep Learning Face Attributes in the Wild'. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3730–3738. DOI: 10.1109/ICCV.2015.425.

[42] David G. Lowe. 'Distinctive Image Features from Scale-Invariant Keypoints'. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94.

[43] Debbie S Ma, Joshua Correll and Bernd Wittenbrink. 'The Chicago face database: A free stimulus set of faces and norming data'. In: *Behavior research methods* 47.4 (2015), pp. 1122–1135.

[44] Satya Mallick. *Face morph using OpenCV — C++ / Python*. URL: https://learnopencv.com/face-morph-using-opencv-cpp-python/ (visited on 30/11/2022).

[45] Aleix Martinez and Robert Benavente. 'The AR face database'. In: *Cvc technical report* 24 (1998).

[46] *Mask.ID*. The Peng! Collective. URL: https://pen.gg/campaign/mask-id-2/ (visited on 02/02/2023).

[47] JJW Meijer. 'Morphing detection based on regional analysis of local frequency content'. B.S. thesis. University of Twente, 2020.

[48] National Institute of Standards and Technology. *NIST FRVT Morph*. URL: https://pages.nist.gov/frvt/html/frvt_morph.html (visited on 30/11/2022).

[49] Tom Neubert, Christian Kraetzer and Jana Dittmann. 'A face morphing detection concept with a frequency and a spatial domain feature space for images on eMRTD'. In: *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*. 2019, pp. 95–100.

[50] Tom Neubert et al. 'Extended StirTrace benchmarking of biometric and forensic qualities of morphed face images'. In: *IET Biometrics* 7.4 (2018), pp. 325–332.

[51] Javier Ortega-Garcia et al. 'The Multiscenario Multienvironment BioSecure Multimodal Database (BMDB)'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.6 (2010), pp. 1097–1111. DOI: 10.1109/TPAMI.2009.76.

[52] Omkar M Parkhi, Andrea Vedaldi and Andrew Zisserman. 'Deep face recognition'. In: British Machine Vision Association, 2015, pp. 1–12.

[53] P Jonathon 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, pp. 947–954.

[54] P Jonathon Phillips et al. 'The FERET database and evaluation procedure for face-recognition algorithms'. In: *Image and vision computing* 16.5 (1998), pp. 295–306.

[55] Alyssa Quek. *FaceMorpher morphing algorithm*. URL: https://github.com/alyssaq/face_morpher (visited on 30/11/2022).

[56] R Raghavendra, Kiran Bylappa Raja and Christoph Busch. 'Exploring the Usefulness of Light Field Cameras for Biometrics: An Empirical Study on Face and Iris Recognition'. In: *IEEE Transactions on Information Forensics and Security* 11.5 (2016), pp. 922–936. DOI: 10.1109/TIFS.2015.2512559.

[57] R Raghavendra et al. 'Transferable Deep-CNN Features for Detecting Digital and Print-Scanned Morphed Face Images'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1822–1830. DOI: 10.1109/CVPRW.2017.228.

[58] Kiran Raja, Sushma Venkatesh, Christoph 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.

[59] Kiran Raja et al. 'Morphing Attack Detection-Database, Evaluation Platform, and Benchmarking'. In: *IEEE transactions on information forensics and security* 16 (2020), pp. 4336–4351.

[60] Raghavendra Ramachandra et al. 'Detecting face morphing attacks with collaborative representation of steerable features'. In: *Proceedings of 3rd international conference on computer vision and image processing*. Springer. 2020, pp. 255–265.

[61] Eklavya Sarkar et al. 'Are GAN-based morphs threatening face recognition?' In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 2959–2963.

[62] Eklavya Sarkar et al. 'Vulnerability analysis of face morphing attacks from landmarks and generative adversarial networks'. In: *arXiv preprint arXiv: 2012.05344* (2020).

[63] Ulrich Scherhag. 'Face Morphing and Morphing Attack Detection'. PhD thesis. Technical University of Darmstadt, Germany, 2021.

[64] Ulrich Scherhag, Christian Rathgeb and Christoph Busch. 'Morph Detection from Single Face Image: A Multi-Algorithm Fusion Approach'. In: *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications*. ICBEA '18. Association for Computing Machinery, 2018, pp. 6–12. DOI: 10.1145/3230820.3230822.

[65] Ulrich Scherhag 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.

[66] Ulrich Scherhag et al. 'Deep Face Representations for Differential Morphing Attack Detection'. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3625–3639. DOI: 10.1109/TIFS.2020.2994750.

[67] Ulrich Scherhag et al. 'Detecting Morphed Face Images Using Facial Landmarks'. In: *Image and Signal Processing*. Ed. by Alamin Mansouri et al. Springer International Publishing, 2018, pp. 444–452. ISBN: 978-3-319-94211-7. DOI: 10.1007/978-3-319-94211-7_48.

[68] Ulrich Scherhag et al. 'Detection of face morphing attacks based on PRNU analysis'. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.4 (2019), pp. 302–317. DOI: 10.1109/TBIOM.2019.2942395.

[69] Ulrich Scherhag et al. 'Face morph detection for unknown morphing algorithms and image sources: a multi-scale block local binary pattern fusion approach'. In: *IET Biometrics* 9.6 (2020), pp. 278–289.

[70] Ulrich Scherhag et al. 'Face recognition systems under morphing attacks: A survey'. In: *IEEE Access* 7 (2019), pp. 23012–23026.

[71] Florian Schroff, Dmitry Kalenichenko and James Philbin. 'FaceNet: A unified embedding for face recognition and clustering'. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[72] Sefik Ilkin Serengil and Alper Ozpinar. 'LightFace: A Hybrid Deep Face Recognition Framework'. In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2020, pp. 23–27. DOI: 10.1109/ASYU50717.2020.9259802.

[73] Chang Shu, Xiaoqing Ding and Chi Fang. 'Histogram of the Oriented Gradient for Face Recognition'. In: *Tsinghua Science & Technology* 16.2 (2011), pp. 216–224. ISSN: 1007-0214. DOI: 10.1016/S1007-0214(11)70032-3.

[74] Karen Simonyan and Andrew Zisserman. 'Very deep convolutional networks for large-scale image recognition'. In: *arXiv preprint arXiv:1409.1556* (2014).

[75] Douglas B Smythe. 'A two-pass mesh warping algorithm for object transformation and image interpolation'. In: *Rapport technique* 1030 (1990), p. 31.

[76] Luuk Spreeuwers, Maikel Schils and Raymond Veldhuis. 'Towards robust evaluation of face morphing detection'. In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, pp. 1027–1031.

[77] Christian Szegedy et al. 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[78] Raphael Thelen and Judith Horchert. 'Aktivisten schmuggeln Fotomontage in Reisepass'. In: *Der Spiegel* (22nd Sept. 2018). URL: https://www.spiegel.de/netzwelt/netzpolitik/biometrie-im-reisepass-peng-kollektiv-schmuggelt-fotomontage-in-ausweis-a-1229418.html (visited on 02/02/2023).

[79] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. 'A new ranking method for principal components analysis and its application to face image analysis'. In: *Image and Vision Computing* 28.6 (2010), pp. 902–913. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2009.11.005.

[80] Laurens Van der Maaten and Geoffrey Hinton. 'Visualizing data using t-SNE'. In: *Journal of machine learning research* 9.11 (2008).

[81] Paul Viola and Michael J Jones. 'Robust real-time face detection'. In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[82] George Wolberg. *Digital Image Warping*. Systems. IEEE Computer Society Press, July 1990.

[83] A Wolf. 'ICAO: Portrait Quality (Reference Facial Images for MRTD), Version 1.0. Standard'. In: *International Civil Aviation Organization* (2018).

[84] xiberpix. *Sqirlz morphing algorithm*. URL: `https://sqirlz-morph.it.uptodown.com/windows` (visited on 30/11/2022).

[85] Le-Bing Zhang, Fei Peng and Min 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. DOI: `10.1109/ICME.2018.8486607`.

[86] Kaipeng Zhang et al. 'Joint face detection and alignment using multitask cascaded convolutional networks'. In: *IEEE signal processing letters* 23.10 (2016), pp. 1499–1503.

[87] Xu Zhang, Svebor Karaman and Shih-Fu Chang. 'Detecting and simulating artifacts in GAN fake images'. In: *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2019, pp. 1–6.