IET Journals
The Institution of Engineering and Technology

Face Morph Detection for Unknown Morphing Algorithms and Image Sources: A Multi-Scale Block Local Binary Pattern Fusion Approach

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

U. Scherhag[1] *J. Kunze*[1] *C. Rathgeb*[1] *C. Busch*[1]

[1] *da/sec – Biometrics and Internet Security Research Group, Hochschule Darmstadt, Germany*
** E-mail: ulrich.scherhag@h-da.de*

Abstract: The vulnerability of face recognition systems against so-called morphing attacks has been revealed in the past years. Recently, different kinds of morphing attack detection approaches have been proposed. However, the vast majority of published results has been obtained from rather constrained experimental setups. In particular, most investigations do not consider variations in morphing techniques, image sources, and image post-processing. Hence, reported performance rates can not be maintained in realistic scenarios, as the NIST FRVT MORPH performance evaluation showed. In this work, existing algorithms are benchmarked on a new, more realistic database. This database consists of two different data sets, from which morphs were created using four different morphing algorithms. In addition, the database contains four different post-processings (including print-scan transformation and JPEG2000 compression), which simulate the processing pipeline of the creation of an electronic travel document. Further, a new morphing attack detection method based on a fusion of different configurations of Multi-scale Block Local Binary Patterns (MB-LBP) on an image divided into multiple cells is presented. MB-LBP features are extracted from face images using various block sizes and cell divisions. For each configuration SVM classifiers are separately trained to distinguish between morphed and bona fide face images. The proposed score-level fusion of a maximum number of 18 different configurations is shown to significantly improve the robustness of the resulting morphing attack detection scheme, yielding an average performance between 2.26% and 8.52% in terms of Detection Equal Error Rate (D-EER), depending on the applied post-processing.

1 Introduction

Image manipulation techniques can be applied to substantially change the appearance of face images and hence negatively affect the recognition accuracy and security of face recognition systems. Face alteration methods include replacement or reenactment [1, 2], which are frequently referred to as "face swapping" or "deep-fakes", retouching [3, 4] as well as morphing [5, 6]. Morphing techniques can be used to create artificial face images that resemble the biometric information of two (or more) subjects in the image and feature domain. Usually, the morphing process comprises the definition of corresponding landmarks, averaging, triangulation, warping, and alpha-blending [5]. Alternatively, morphs might as well be created using Generative Adversarial Networks (GANs) [7]. An example of a morphed facial image is shown in Fig. 1b. With high probability, the morphed facial image is successfully verified against probe samples from both subjects contributing to the morph using state-of-the-art Face Recognition Systems (FRS). This means that if a morphed facial image is somehow stored as a reference in the database of a FRS or in the chip of an electronic travel document, both subjects involved can successfully be verified against this manipulated reference. Morphed facial images thus pose a serious threat to FRSs as the basic principle of biometrics, i.e. the unambiguous link between biometric data and the subject, is broken.

In 2014, Ferrara et al. [37] were the first to thoroughly investigate the vulnerability of a commercial FRS against *face morphing attacks*. So far, a considerable amount of *morphing attack detection* mechanisms has been published. For a comprehensive survey the reader is referred to [5]. Proposed approaches can be categorized according to the morphing attack detection scenario. In the *no-reference* morphing attack detection scenario, the detector processes a single image, e.g. an image that is presented in a passport application procedure (this scenario is also referred to as single image morphing attack detection or forensic morphing attack detection). On the contrary, in the *differential* morphing attack detection scenario, a trusted live capture from an authentication attempt serves as additional source of information for the morph detector, e.g. during authentication at an Automatic Border Control (ABC) gate (this scenario is also referred to as image pair-based morphing attack detection). Note that all information extracted by no-reference morph detectors might as well be leveraged within this scenario [17].

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

Fig. 1: Example for a morphed face image (b) of subject 1 (a) and subject 2 (c); the morph was created using FantaMorph.

In this work, focus is put on the more challenging no-reference scenario. A comprehensive evaluation on two different face databases using four morphing algorithms and four post-processing methods is conducted. It is shown that a fusion of multiple configurations of Multi-scale Block LBP (MB-LBP) improves the performance as well as the robustness of the morphing attack detection system. Further, the proposed fusion-based scheme that combines the complementary information extracted from various scales outperforms diverse published approaches. Moreover, as opposed to existing works, it is shown that morphing attack detection

Table 1 Overview of most relevant no-reference face morphing attack detection algorithms (adapted from [5]).

Ref.	Approach	Category	Morphing method	Source face database	Post-processing	Remarks
[8]	BSIF with SVM	texture descriptors	GIMP/GAP	in-house	–	–
[9]	BSIF with SVM	texture descriptors	GIMP/GAP	in-house	print and scan	fixed database of [8]
[10]	Multi-channel-LBP with Pro-CRC	texture descriptors	OpenCV	FRGCv2	print and scan	–
[11]	Multi-channel-LBP with SRKDA	texture descriptors	[12]	[12]	print and scan	–
[13]	WLMP with SVM	texture descriptors	Snapchat	in-house	–	–
[14, 15]	ULBP and RIPS with KNN	texture descriptors	[16]	Utrecht	–	–
[17]	BSIF with SVM	texture descriptors	triangulation + blending	FRGCv2	–	–
[18]	Score-level fusion of general purpose image descriptors	texture descriptors	triangulation + blending	FRGCv2	–	–
[19]	HOG with SVM	texture descriptors	triangulation + blending	FRGCv2, FERET, ARface	–	cross-database performance evaluation
[20]	LBP with SVM	texture descriptors	triangulation + blending	FRGCv2, FERET	–	cross-database performance evaluation
[7]	LBP with SVM	texture descriptors	MorGan [7]	CelebA	–	–
[21]	High-Dim. LBP with SVM	texture descriptors	triangulation + blending + swapping	Multi-PIE	–	–
[22]	Image degradation	digital forensics	triangulation + blending (+ swapping)	in-house, Utrecht	–	–
[23–25]	PRNU analysis	digital forensics	triangulation + blending	FRGCv2	hist. equalization, scaling, sharpening	–
[26]	PRNU analysis	digital forensics	triangulation + blending MorGan [7]	CelebA	–	–
[27]	SPN analysis	digital forensics	triangulation + blending (+ swapping)	Utrecht, FEI	–	–
[16]	Double-compression artefacts analysis	digital forensics	triangulation + blending (+ swapping)	Utrecht, FEI	–	–
[28]	Double-compression artefacts analysis	digital forensics	[16]	Utrecht, FEI	–	–
[29]	Reflection analysis	digital forensics	triangulation + blending (+ swapping)	in-house	–	–
[12]	Luminance component and steerable pyramid with ProCRC	digital forensics	triangulation + blending (+ swapping)	[10] extended	print and scan	–
[30]	Image quality features with SVM	digital forensics	GAN generated Morphs	VidTIMIT	–	–
[31]	VGG19 and AlexNet with ProCRC	deep learning	[9]	in-house	print and scan	–
[32]	VGG19, GoogLeNet, AlexNet	deep learning	triangulation + blending (+ swapping)	in-house	–	–
[33]	VGG19	deep learning	triangulation + blending (+ swapping)	BU-4DFE, CFD, FEI, FERET, PUT, scFace, Utrecht, in-house	motion blur, Gaussian blur, salt-and-pepper noise, Gaussian noise	trained on all combinations (no unseen attack classes)
[34]	OpenFace NN4.SMALL2 and LBP with SVM	deep learning and texture descriptors	[35]	CelebA	–	candidate selection presented in [35]
[36]	VGG19 with SVM	deep learning	triangulation + blending (+ swapping)	FRGCv2, FERET, ARface, Biometix	print and scan	–

remains a challenging task in real-world scenarios where the image source and/or the algorithm used to morph face images is unknown to the detection system.

The remainder of the manuscript is organized as follows: In Sec. 2, the related work is briefly revisited. Subsequently, the used image databases are summarized in Sec. 3. In Sec. 4, the proposed system is described in detail. An in-depth evaluation is presented in Sec. 5. Finally, a conclusion is given in Sec. 6.

2 Related Work

In general, no-reference face morphing attack detectors can be divided into three algorithm classes which utilize either (1) *texture descriptors*, (2) *digital forensics*, or (3) *deep learning*. Most relevant published approaches and their properties are listed in Table 1.

During the morphing process various artefacts are created which can be detected by analyzing the texture. Due to the averaging of two images, the resulting morph is smoothed, e.g. the skin textures will loose their sharpness. Furthermore, ghost artefacts or half-shade effects occur if the two morphed images are not aligned correctly and if there are too few or incorrectly positioned landmarks. Especially in the area of the pupils and the nostrils these artefacts occur more frequently. Other artefacts detectable by texture descriptors are distorted corners and offset image areas. In several publications the use of common texture descriptors, e.g. Local Binary Patterns (LBP) [38] or Binarized Statistical Image Features (BSIF) [39], has already been demonstrated [7–9, 17, 20]. An extension of these algorithms to several color channels [10, 11] or higher dimensions [21] can lead to further improvements. Other texture descriptors such as Unified Local Binary Patterns (ULBP) [14, 15] or Weighted Local Magnitude Patterns (WLMP) [13] have also been tested.

The distortion and blending during the morphing process has an influence on the high-frequency information of the image. These changes can be analyzed by image forensics-based detection methods. For example, it has been shown that morphs can be detected by analysing Photo Response Non-Uniformity (PRNU) [23–26] or sensor pattern noise (SPN) [27]. Moreover, the quality of the images is reduced by editing and saving them in the morphing process. Under the assumption that the quality of morphed images is always lower than those of bona fide images, image quality can be used for morph detection. This can be done by either analyzing intentional degradation of the image in question [22] or by using several quality features in combination with a classifier [30]. Under the assumption that the images are stored in a lossy compression format before and after morphing, it is possible to detect morphs by analyzing double compression artefacts [16, 28]. Furthermore, the images can be examined for inconsistencies, for example for non-natural lighting conditions or color values [29].

Fig. 2: Examples for a morphed face images from all four algorithms (resized). From left to right: Subject 1, FaceFusion morph, FaceMorpher morph, OpenCV morph, UBO-Morpher morph, and Subject 2

Table 2 Number of subjects, bona fide and morphed face images (per morphing algorithm) of used datasets. "F" and "M" indicate female and male subjects, respectively.

Database	Subjects	Face images	
		bona fide	morphed
FRGCv2	533 (231 F, 302 M)	1441	964
FERET	529 (200 F, 329 M)	622	529

The third class of no-reference algorithms are based on deep learning. Deep learning-based feature extractors offer the advantage that they can theoretically learn to detect any artefact present in the training set. This, however, carries the risk of over-fitting to artefacts, which only occur with the morph algorithms used for training and are therefore not generally valid. One possibility is the training or transfer learning of a network for the detection of morphs [32, 33]. Another possibility is the use of pre-trained neural networks for feature extraction in combination with a classifier (e.g. SVM) [31, 36]. Deep features can also be combined with other features (e.g. LBP) [34].

While the majority of morph detection approaches report practical detection error rates, these are commonly evaluated on a dataset of bona fide and morphed face images which are extracted from a single (in-house) face database and created by a single morphing algorithm. It was shown, that variations in dataset [19], morphing process, and post-processing (e.g. print and scan [9]) might negatively influence the performance of the morph detection algorithms. This has also been confirmed in the Face Recognition Vendor Test (FRVT) MORPH conducted by the National Institute of Technology (NIST) [40]. In [18] a fusion of multiple algorithms was proposed, as it might improve the detection performance of no-reference algorithms. Even the fusion of different configurations of the same algorithm were found to be beneficial.

3 Database

The results of this work were obtained based on subsets of the FRGCv2 [41] and FERET [42] face image databases. From these databases, potential reference images meeting International Civil Aviation Organization (ICAO) passport photo quality standards [43] are selected. From the pre-selected images, image pairs are created for the morphing process. Where possible, different images are used for morphing and as bona fide samples. However, for some subjects there are not enough samples, so the same image is used in both subsets (morphed and bona fide). The number of used subjects, bona fide images as well as the number of created morphs are given in Table 2.

3.1 Morphing

Different morphing algorithms produce morphs with different artefacts. For a comprehensive evaluation a database with different morphing algorithms is therefore necessary to ensure that the morph attack detection algorithms have not stiffened to algorithm-specific artefacts. For this purpose, four morphing algorithms were used to ensure a large variation of morphs, examples are shown in Fig. 2 with equal contribution of both subjects:

1. **FaceFusion**[*], a proprietary morphing algorithm. Due to the inaccessible source code it is not possible to determine in which way the morphs are generated. It can be seen, that after the morphing process parts of the first subject are blended over the morph to hide artefacts (eyes, nostrils, outer facial region). The created morphs have a high quality and low to no visible artefacts.
2. **FaceMorpher**[†], an open-source implementation using Python. In the used version the algorithm applies STASM for landmark detection. Delaunay triangles are formed from the landmarks, which are distorted and blended. The area outside the landmarks is averaged. The generated morphs show strong artefacts in particular in the area of neck and hair.
3. **OpenCV**, a self-made morphing algorithm based on the tutorial "Face Morph Using OpenCV"[‡]. This algorithm works similar to FaceMorpher. Important differences between the algorithms are that for landmark recognition Dlib is used instead of STASM and that for this algorithm landmarks are positioned at the edge of the image, which are also used to create the morphs. Thus, in contrast to FaceMorpher, the edge does not consist of an averaged image, but like the rest of the image, of morphed triangles. However, also in this version strong artefacts outside the facial area can be observed, which is mainly due to missing landmarks.
4. **UBO-Morpher**, the morphing tool of University of Bologna, as used e.g. in [44]. This algorithm receives two input images as well as the corresponding landmarks. Dlib landmarks were used for this morphing tool. The morphs are generated by triangulation, averaging and blending. To avoid the artefacts in the area outside the face, the morphed face is copied to the background of one of the original images. Even if the colors are adjusted, visible edges may appear at the transitions.

In order to be able to conduct a fair benchmark in our experiments, the same combination of morphed face images was created for each of the listed algorithms.

3.2 Post-processing

In addition to the considered ICAO compliance, various post-processings of the images must also be taken into account, since images of the database aim at imitating real-world scenario of the application process of an electronic travel document. In many countries the images are down-scaled, e.g. to 360×480 pixels, and compressed, e.g. to 15kB using JPEG2000, prior to storing them on the chip of an electronic travel document, e.g. an ePassport. In addition, the images can be handed over in printed form by the applicant. It can be assumed that morphs are more easier to be recognized

[*] `www.wearemoment.com/FaceFusion/`
[†] `github.com/alyssaq/face_morpher`
[‡] `www.learnopencv.com/face-morph-using-opencv-cpp-python/`

(a) Resized (b) JPEG2000 (c) Print/Scan (d) Print/Scan and JPEG2000

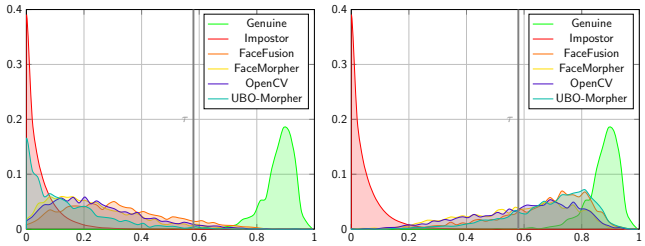
Fig. 3: Comparison of different post-processings (FaceFusion). Zoomed in to reveal artefacts and noise more clearly.

in unprocessed images and that each post-processing step increases the difficulty of reliable detection. In order to cover the realistic scenarios, the following post-processings have been applied:

1. **Resizing (RS):** The resolution of the images is reduced to the minimum inter-eye distance (90px) required by the ICAO guidelines for electronic travel documents [43]. This post-processing corresponds to the scenario that an image is submitted digitally by the applicant. An example is shown in Fig. 3a. This post-processing is applied in advance to all subsequent post-processings described below.
2. **JPEG2000 Compression (J2):** A wavelet-based image compression method that is recommended for electronic travel documents [45]. The setting is selected in a way that a target file size of 15KB is achieved. This post-processing corresponds to the scenario that a digitally submitted image is stored in the chip of the electronic travel document. An example is shown in Fig. 3b.
3. **Printing and Scanning (PS):** The images are first printed with a high quality laser printer (*Fujifilm Frontier 5700R Minlab* on *Fujicolor Crystal Archive Papier Supreme HD Lustre photo paper*) and then scanned with a premium flatbed scanner (*Epson DS-50000*) with 300 dpi. A dust and scratch filter is then applied in order to reduce image noise. This post-processing corresponds to the scenario that an analog image is submitted with the electronic travel document application. An example is shown in Fig. 3c.
4. **Printing, Scanning and JPEG2000 Compression (PS-J2):** A combination of the previous post-processings. The images are first printed and scanned and then compressed using JPEG2000. This post-processing corresponds to the scenario that a analog submitted image is stored in the chip of the electronic travel document. An example is shown in Fig. 3d.

3.3 Validation of Attack Potential

To assure the significance of the following experiments, the attack potential of the created databases is evaluated in a first step. For this purpose, comparison scores for genuine and impostor comparisons, as well as for morphing attacks are determined and the Mated Morph Presentation Match Rate (MMPMR) and the Relative Morph Match Rate (RMMR) defined in [46] is estimated. The FRGCv2 provides probe images showing a significantly higher variance (and therefore higher realism) compared to the probe images contained in the FERET database, thus the validation of the attack potential is limited to the FRGCv2 database. Due to the lower variance of the sample images, the comparisons of the FERET database results in higher comparison scores for genuine and morph attack comparisons, thus the results obtained on FRGCv2 can be considered as a lower limit for the attack potential. The comparison scores were generated using a Commercial-Of-The-Shelf (COTS) FRS. The resulting probability density functions are depicted in Fig. 4. In most publications,



(a) 25/75 Morphs (b) 50/50 Morphs

Fig. 4: Probability Density Functions (PDFs) of comparison scores of genuine, impostor, and morphing attacks for symmetrical and asymmetrical morphs. τ depicts the estimated threshold for a FMR of 0.1%.

Table 3 Vulnerability assessment of COTS FRS.

	MMPMR/RMMR in %			
α	**FaceFusion**	**FaceMorpher**	**OpenCV**	**UBO-Morpher**
0.25	18.8	8.4	9.8	3.0
0.5	79.4	60.1	62.8	81.5

databases with symmetric morphs are used. This means that both subjects are equally contributing to the creation of the morph. However, it is also suggested, e.g. in [44], to assign a lower weight to one subject, in order to increase the chances in the case of a manual control with this subject. For this reason, in addition to the probability density functions of symmetrical morphs in Fig. 4b, the distributions of asymmetrical morphs with a weighting of 25% and 75% ($\alpha = 0.25$) are shown in Fig. 4a, the corresponding MMPMR and RMMR are listed in Table 3. Since the FRS maintains a zero FNMR at the considered FMR of 0.1% the MMPMR is equal to the RMMR. However, it is evident that the asymmetric morphs, regardless of the applied morphing algorithm, have no attack potential for the used FRS. This behaviour is reinforced by the realistic variance of the probe images used. As a consequence, only symmetrical morphs are considered in this paper.

4 Proposed System

The proposed system, which is depicted in Fig. 5, comprises three key modules, (1) *MB-LBP extraction*, (2) *cell division* and (3) *training and score-level fusion*; in the following subsections, all modules are described in detail. To avoid algorithm overfitting on avoidable

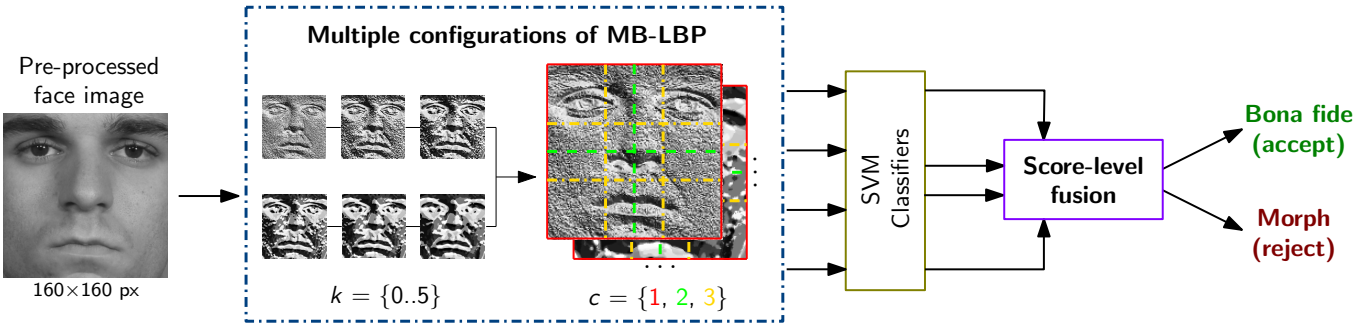


Fig. 5: Overview of the proposed multiple configuration MB-LBP fusion approach with division into multiple cells to detect morphed facial images. k is the parameter for the MB-LBP block size and c the parameter for the cell division.

artefacts, e.g. ghost-artefacts in hair regions, the image is cropped to a size of 320×320 pixels using predefined offsets, whereby the image area showing the face is cut out. Finally, the cropped face part is converted to a grayscale image.

4.1 Multi-scale Block LBP

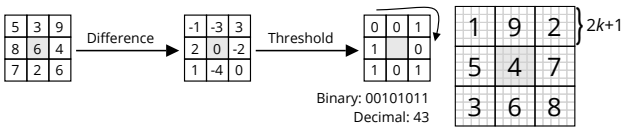
LBP has been found to be a powerful feature for texture classification. Specifically, LBP has been shown to be suitable for detecting morphed face images in no-reference scenarios [18]. The distortions of the images introduced by the morphing process are changing the texture of the images in a way, that can be detected in an LBP-histogram. Further, the images are averaged during the blending, which smooths the resulting morph, leading to less sharp edges, which are reflected in an LBP-histogram, too. In addition, the morphing process might introduce minor artefacts to the image [46]. As LBP is designed for the representation of surface properties, these artefacts can be represented in the LBP-histogram as well and can be utilized to detect morphed face images.

The original LBP operator labels the pixels of an image by thresholding the 3×3-neighborhood of each pixel with the center value and considering the result as a binary string or a decimal number. Then the histogram of extracted LBP values can be used as a texture descriptor. MB-LBP [47] is an extension to the basic LBP, with respect to neighborhoods of different sizes. In MB-LBP, the comparison operator between single pixels in LBP is replaced with the comparison between average pixel intensities of sub-regions. Each sub-region is a square block containing neighboring pixels. In each sub-region, the average sum of pixel intensities is computed. These average sums are then thresholded by that of the center block. The whole filter is composed of 9 blocks (center block and 8 neighbouring blocks) of size $(2k+1)\times(2k+1)$ pixels. If a higher value for k is selected, details are lost while robustness increases [47]. An example of the basic LBP and the MB-LBP operator is shown in Fig. 6. In order to be able to compute the LBP blocks in the peripheral regions, padding border lines and columns are added to the image in advance which replicates the outer pixel values.

4.2 MB-LBP feature extraction over multiple cells

Even if the performance of LBP in constrained scenarios is promising, the detection performance of LBP highly degrades when the face images are post-processed, e.g. by printing and scanning. Further, it was observed, that smaller blocks show a higher performance on single databases, but larger blocks are more robust in a cross-database analysis [19]. Scherhag et al. [18] have shown that a fusion of two LBP configurations might lead to an increased performance and robustness of the algorithm.

After the computation of the MB-LBP values, the resulting image is divided into $c \times c$ cells. For each cell a histogram is calculated, the individual histograms are concatenated to a longer MB-LBP feature vector. As c increases, so does the number of concatenated histograms and thus the size of the feature vector. With that comes the benefit of retaining more local information.



(a) basic LBP (b) MB-LBP

Fig. 6: The basic LBP Operator and the MB-LBP operator with k=2.

Thus, at feature-extraction, the MB-LBP feature extraction is applied to the post-processed image in different configurations. The configurations consist of the possible combinations resulting from the values for k and c. Values from 0 to 5 are selected for k, since too much information is lost with even larger values. The picture is divided into a maximum of 3×3 cells ($c = \{1, 2, 3\}$), otherwise the ratio between the patch size and the cell size is disproportionate. This results in $6 \times 3 = 18$ possible configurations.

4.3 Training and score-level fusion

To distinguish between bona fide and morphed face images one support vector machine (SVM) is trained per configurations of k and c. The default hyper parameters of the scikit-learn implementation of linear kernel SVM* are used (C=1.0, gamma=(n_features × variance)$^{-1}$). For a given face image each SVM generates a normalized attack detection score in the range [0,1].

In the fusion stage a sum-rule score-level fusion is applied to the scores of the different classifiers. The number of fused algorithms ranges from 1 (no fusion) to the total number of MB-LBP configurations and cell divisions, i.e. 18. Considering all possible combinations, this results in a quantity of

$$\sum_{n=1}^{18} \binom{18}{n} = 262,143$$

fusions. Despite this large amount of possible MB-LBP configurations, it is expected that the maximum number of configurations reveals competitive detection performance, as will be shown in experiments.

5 Experiments

In the following section, the experimental setup as well as the evaluation of the experiments are described, including a discussion of the observed results. Performance evaluations are conducted based on the database described in Sec. 3.

*https://scikit-learn.org/stable/modules/svm.html

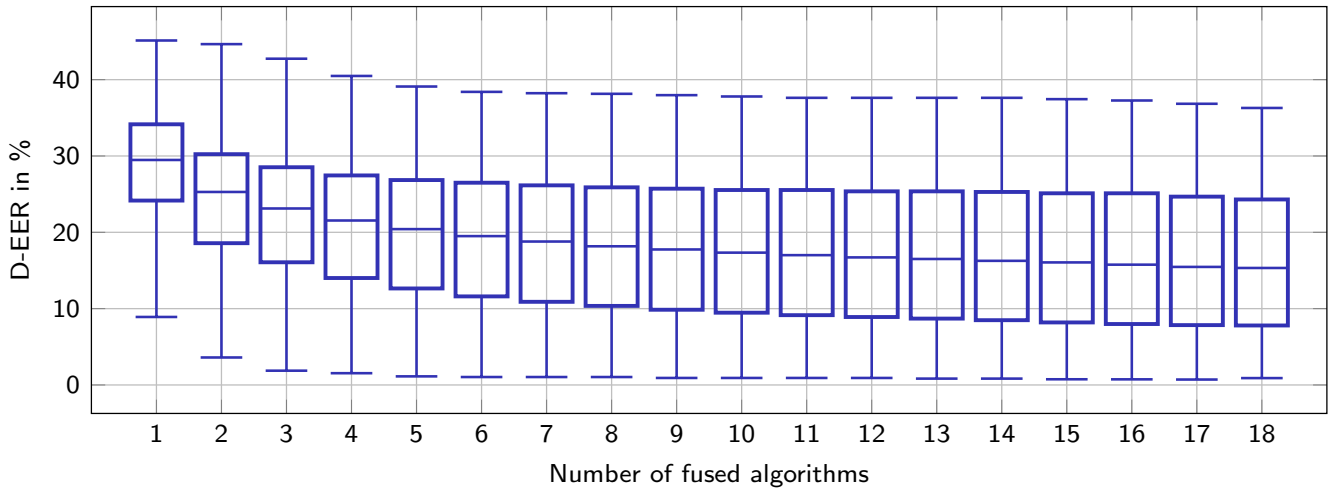


Fig. 7: Box plot over the distribution of D-EERs of all possible fusion combinations of a fixed number of algorithms.

Table 4 Average D-EER for different numbers of fused MB-LBP configurations.

	Number of fused MB-LBP configurations																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
D-EER	29.5	25.3	23.1	21.5	20.4	19.5	18.8	18.2	17.8	17.3	17.0	16.7	16.5	16.3	16.1	15.8	15.5	15.3

5.1 Morph detection performance evaluation

For the performance evaluation of the described algorithm, the SVM classifiers are each trained on one post-processing and one morphing algorithm at a time using FERET database. The evaluation is performed on FRGCv2 database and all other morphing algorithms, resulting in 12 combinations per post-processing and 48 combinations in total.

The performance of the detection algorithms is reported using the Detection Equal Error Rate (D-EER), i.e. the operating point where the proportion of attack presentations incorrectly classified as bona fide presentations (APCER) is as high as the proportion of bona fide presentations incorrectly classified as presentation attack (BPCER). For APCER and BPCER the definitions of ISO IEC 30107-3 [48] are used:

Table 5 D-EER (in %) of single MB-LBP algorithms and the fused approach. The D-EER of an algorithm trained on all available attack types is highlighted in italic.

Train		**Test**		**Best Single Detection-Alg. Performance**			**Fused Detection-Alg. Performance**	
Database	**Morph-Alg.**	**Database**	**Morph-Alg.**	**Configuration**	**One Morph-Alg.**	**All Morph-Alg.**	**One Morph-Alg.**	**All Morph-Alg.**
FERET	FaceFusion	FRGC	FaceMorpher	k: 1 - c: 3	6.3	31.0	7.9	10.8
			OpenCV	k: 1 - c: 2	19.4	35.7	15.3	
			UBO-Morpher	k: 1 - c: 3	15.6	31.0	12.0	
	FaceMorpher		FaceFusion	k: 1 - c: 3	20.5	31.0	17.8	
			OpenCV	k: 0 - c: 3	8.1	28.0	14.5	
			UBO-Morpher	k: 0 - c: 3	9.0	28.0	11.0	
	OpenCV		FaceFusion	k: 1 - c: 3	19.9	31.0	14.3	
			FaceMorpher	k: 0 - c: 1	0.3	22.0	6.3	
			UBO-Morpher	k: 0 - c: 3	11.2	28.0	10.0	
	UBO-Morpher		FaceFusion	k: 0 - c: 2	15.1	32.3	9.4	
			FaceMorpher	k: 0 - c: 3	1.9	28.0	2.2	
			OpenCV	k: 0 - c: 2	11.4	32.3	9.1	
	All		*FaceFusion*	-	-	-	*8.1*	*5.7*
			FaceMorpher	-	-	-	*1.1*	
			OpenCV	-	-	-	*6.7*	
			UBO-Morpher	-	-	-	*6.9*	

Table 6 Morph-detection performance in terms of D-EER (in %) of state-of-the-art algorithms.

		Train: FaceFusion. Test:			Train: FaceMorpher. Test:			Train: OpenCV. Test:			Train: UBO-Morpher. Test:			
Algorithm	**Post-Processing**	**FaceMorpher**	**OpenCV**	**UBO-Morpher**	**FaceFusion**	**OpenCV**	**UBO-Morpher**	**FaceFusion**	**FaceMorpher**	**UBO-Morpher**	**FaceFusion**	**FaceMorpher**	**OpenCV**	**Average**
BSIF	RS	25.21	21.77	23.67	26.13	16.79	20.33	22.84	12.22	19.2	21.66	16.48	16.27	**20.21**
	J2	26.44	26.64	20.64	17.09	18.07	18.48	18.48	18.69	18.89	17.4	21.56	23.46	**20.49**
	PS	16.89	12.01	14.84	10.68	11.19	12.78	12.22	15.25	14.53	10.99	12.99	10.99	**12.95**
	PS-J2	25.31	24.18	20.74	22.54	24.08	24.28	21.87	23.46	23.97	18.07	23.36	21.36	**22.77**
BSIF 4×4	RS	16.79	14.73	15.86	20.43	12.22	15.04	17.66	8.42	13.91	17.97	11.6	12.78	**14.78**
	J2	26.13	25.21	19.1	17.76	17.66	17.2	17.09	19.4	17.3	16.58	21.56	22.28	**19.77**
	PS	14.01	12.58	13.6	12.12	12.22	12.88	12.99	14.63	14.01	13.3	14.01	12.88	**13.27**
	PS-J2	25.21	22.54	21.46	22.95	24.08	23.15	22.84	24.59	21.87	19.92	25.31	22.28	**23.02**
ArcFace	RS	30.7	30.8	31.83	33.93	29.57	35.47	32.34	29.16	32.8	32.6	32.49	31.42	**31.93**
	J2	31.42	30.29	31.93	34.34	29.67	35.58	32.91	29.16	33.62	33.01	32.34	30.9	**32.10**
	PS	32.49	32.03	33.11	34.45	30.29	36.91	33.52	28.44	34.34	31.83	31.73	30.6	**32.48**
	PS-J2	32.91	31.42	33.52	35.27	30.08	37.42	32.7	28.03	34.14	32.14	32.6	30.08	**32.53**
LBP	RS	16.99	19.1	20.33	27.26	18.38	20.53	26.23	10.78	19.92	24.69	12.22	17.3	**19.48**
	J2	32.34	34.45	25.72	23.67	23.05	22.95	24.59	26.44	24.08	21.97	27.46	30.18	**26.41**
	PS	17.3	17.51	19.3	15.97	15.55	16.99	17.3	16.89	17.86	15.45	14.22	14.94	**16.61**
	PS-J2	29.47	27.52	24.79	26.23	26.75	26.75	25	26.44	26.03	22.95	27.93	25.92	**26.32**
LBP 4×4	RS	10.88	14.12	16.38	23.56	14.63	17.86	21.05	7.44	16.48	19.92	8.62	13.81	**15.40**
	J2	30.49	31.42	24.08	20.43	22.18	20.64	22.84	24.38	21.97	21.97	26.85	29.26	**24.71**
	PS	15.66	15.97	17.09	15.35	14.42	15.14	17.09	15.45	15.86	15.97	12.99	14.73	**15.48**
	PS-J2	28.44	27.16	24.79	25.41	25.72	24.79	24.38	25.92	24.08	22.54	26.23	23.25	**25.23**
MB-LBP	RS	2.05	10.88	9.45	2.72	16.27	13.19	8.73	13.19	14.01	6.21	2.93	12.01	**9.30**
	J2	2.26	17.30	14.63	17.66	30.18	27.26	15.25	20.23	18.99	12.32	4.88	20.23	**16.77**
	PS	5.18	9.55	9.75	3.34	13.30	12.32	2.72	2.05	6.42	2.46	0.72	10.58	**6.53**
	PS-J2	2.93	29.88	17.40	37.32	29.47	24.79	23.67	7.24	12.58	25.82	3.75	16.89	**19.31**

APCER: proportion of attack presentations incorrectly classified as bona fide presentations in a specific scenario

BPCER: proportion of bona fide presentations incorrectly classified as presentation attacks in a specific scenario

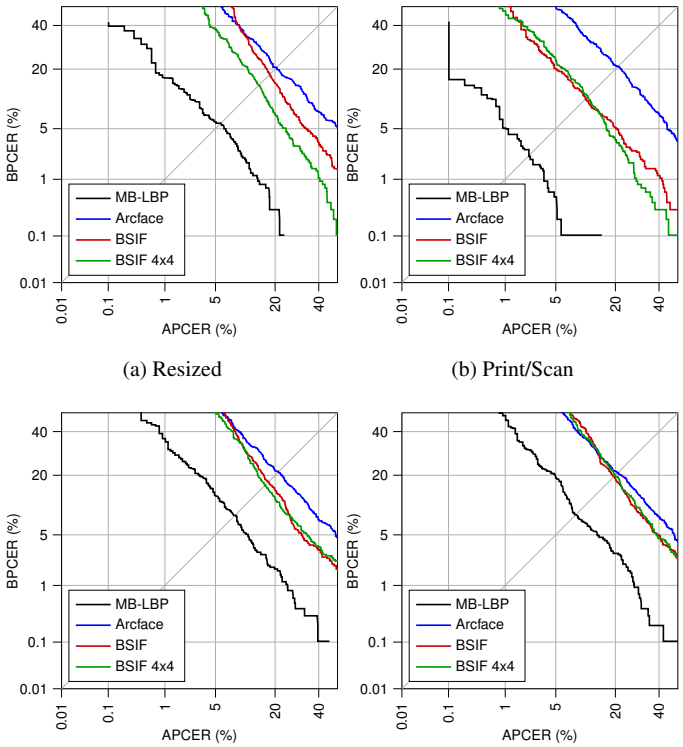
In a preliminary analysis, all possible MB-LBP configurations with different cell divisions described in Sec. 4.2 were trained and tested on images that were not post-processed. The best configuration and the corresponding error rates are listed in Table 5. It is apparent, that a subdivision into more cells ($c = 3$) is preferred. However, no single configuration is equally suitable for all morphing algorithms and databases. E.g. MB-LBP with $k = 0$ and $c = 3$ cells can reach an D-EER as low as 1.9% detecting FRGCv2 FaceMorpher morphs if trained on FERET database and images created by the UBO-Morpher algorithm, but overall this configuration yields an D-EER of only 28.0%.

In order to obtain a more robust morphing attack detection algorithm, multiple MB-LBP configurations can be fused as described in Sec. 4.3. In Fig. 7, a box plot over the distribution of D-EERs of all possible fusion combinations of a fixed number of algorithms is depicted, the corresponding average D-EER per number of fused combinations is listed in Table 4. The maximum number of algorithms to fuse is limited, since the algorithm described in Sec. 4.2 allows for 18 different MB-LBP configurations. With an increasing number of fused algorithms, the median of the D-EERs is lowered, as well as the upper quartile and whisker. Thus, in the remainder of this manuscript, only the fusion of all 18 configurations is considered.

In Table 5 in the two rightmost columns the performance of the fused algorithm is shown. For database and morph algorithms that are easily detectable by single algorithms the fused algorithm performs good as well. For subsets that are harder to detect, the performance of the fused algorithm drops, but in general they are more robust than single algorithms.

5.1.1 Comparison to state-of-the-art algorithms: The morph attack detection performance of common state-of-the-art morph detection algorithms is listed in Table 6. Additionally, the corresponding Detection Error Trade-off (DET) curves are depicted in Fig. 8. The algorithms used for comparison comprise of a open-source facial recognition frameworks based on a ResNet deep neural networks (ArcFace [49]) and two texture descriptors, namely LBP [38] and BSIF [39] with patches of size 3×3 and an optional division into 4×4 cells.

As can be seen, the proposed MB-LBP fusion approach clearly outperforms the algorithms used for comparison. Especially the detection of PS processed morph images performs far better with an average D-EER of 2.26%, whereas the best of the other algorithms yields an average D-EER of 12.95%. The same applies to resized morphs which are significantly better detected by the proposed algorithm (5.80%) than by the other algorithms (14.78%). Also the



(a) Resized (b) Print/Scan

(c) JPEG2000 (d) Print/Scan - JPEG2000

Fig. 8: DET-plots of MB-LBP and state-of-the-art algorithms.

challenging JPEG2000 and Print/Scan - JPEG2000 processed morphed images are better detected by the proposed algorithm with an average D-EER of 8.32% and 8.52%, respectively, while the best of the other algorithms yield an average D-EER of 19.77% and 22.77%, respectively.

In all cases, it is the BSIF algorithm in one of the two configurations that comes closest to the performance of MB-LBP. The superiority of texture algorithms over deep learning algorithms in no-reference scenarios can also be observed in Fig. 8 in the DET plots. In all four plots it can be clearly seen that ArcFace ranks last and delivers performance that is largely unsuitable for practical use in a no-reference scenario. Although the performance of ArcFace is poor, it should be noted that the deep learning algorithm respond much less sensitively to image post-processing and are therefore can be considered to be more robust overall.

In the following, it will be analysed how large the influence on the performance of the fused algorithms is when training and testing is done on differently post-processed images. In addition, it will

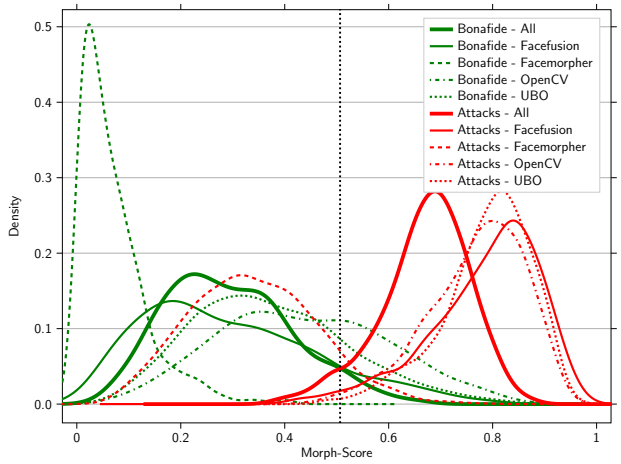


Fig. 9: Kernel Density Estimation - Resized

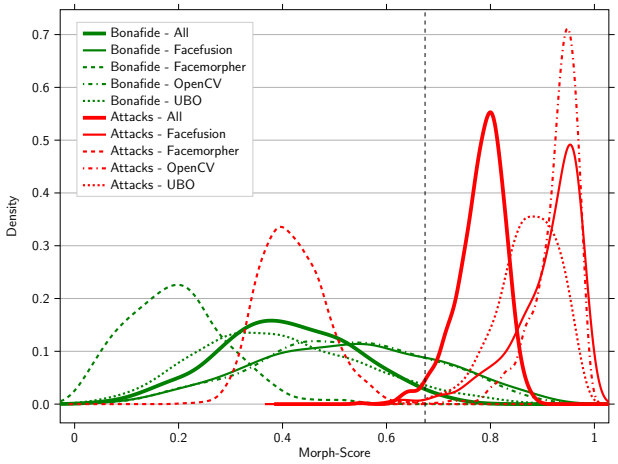


Fig. 10: Kernel Density Estimation - Print/Scan

Table 7 Morph-detection performance when trained/tested on RS images.

	Train	Test	D-EER (in %)	∅	∅
RS	FaceFusion	FaceMorpher	2.05	6.62	5.80
		OpenCV	10.88		
		UBO-Morpher	9.45		
	FaceMorpher	FaceFusion	2.72	7.91	
		OpenCV	16.27		
		UBO-Morpher	13.19		
	OpenCV	FaceFusion	8.73	10.88	
		FaceMorpher	13.19		
		UBO-Morpher	14.01		
	UBO-Morpher	FaceFusion	6.21	5.80	
		FaceMorpher	2.93		
		OpenCV	12.01		
	ALL	*FaceFusion*	*10.78*	10.88	-
		FaceMorpher	*7.34*		
		OpenCV	*14.63*		
		UBO-Morpher	*12.88*		

Table 8 Morph-detection performance when trained/tested on PS images.

	Train	Test	D-EER (in %)	∅	∅
PS	FaceFusion	FaceMorpher	5.18	6.73	2.26
		OpenCV	9.55		
		UBO-Morpher	9.75		
	FaceMorpher	FaceFusion	3.34	7.60	
		OpenCV	13.30		
		UBO-Morpher	12.32		
	OpenCV	FaceFusion	2.72	2.62	
		FaceMorpher	2.05		
		UBO-Morpher	6.42		
	UBO-Morpher	FaceFusion	2.46	2.62	
		FaceMorpher	0.72		
		OpenCV	10.58		
	ALL	*FaceFusion*	*21.46*	2.52	-
		FaceMorpher	*0.31*		
		OpenCV	*4.47*		
		UBO-Morpher	*4.26*		

be shown in which way the performance depends on the choice of the morphing algorithm for training. Therefore in Fig. 9 - 12 Kernel Density Estimation (KDE) plots are given, showing the distribution of attack and bona fide presentations over a range from 0 to 1, with 0 meaning *bona fide* and 1 meaning *attack*. Each pair of thin green and red curves indicates the performance for a morphing algorithm that has been used for training. The thick red and green curve depict the mean performance across all training algorithms, with the D-EER line (dashed vertical line) indicating the threshold that separates the attack and bona fide presentations in average. It should be noted that the EER line differs for the individual post-processings, each representing different application scenarios. It appears, however, that J2 compression seems to be dominating, so that for J2 and PS-J2 an equal error is achieved at the same threshold (0.4).

5.1.2 Resized: The detection performance rates are shown in Table 7. If the resolution of the images is reduced by half, the average D-EER improves by almost 50% compared to the results of the preliminary examination to 5.80%. This can be explained by the fact that the reduction of the resolution and thus the deletion of high-frequency information results in an alignment of the two databases regarding their image structure, such as image noise. In particular for smaller value of k it is more likely, that irrelevant information owed to the image acquisition format are taken into account during training. It can therefore be assumed that training on resized images is more likely to consider information that is actually caused by the morphing process.

As shown in the KDE plot in Fig. 9, the dashed red curve denoting FaceMorpher is far to the left of the EER line, indicating that probably many of the attack images are misclassified as bona fide. In this case training on FaceMorpher could therefore be detrimental to morph attack detection performance.

The performance deteriorates significantly when training is done with all morphing algorithms, but at the same time it becomes more robust.

5.1.3 Print/Scan: The detection performance rates are shown in Table 8. Similar to the resized scenario, the printing and subsequent scanning of the morphed images seems to result in an extensive

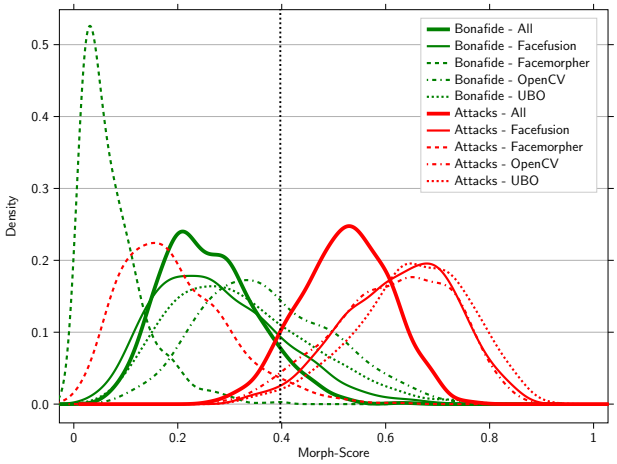


Fig. 11: Kernel Density Estimation - JPEG2000

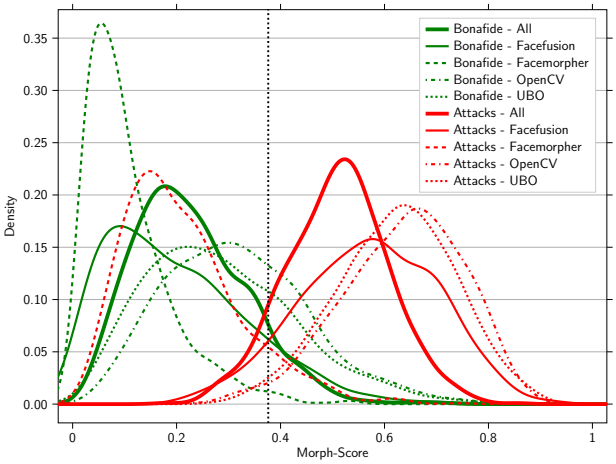


Fig. 12: Kernel Density Estimation - Print/Scan - JPEG2000

Table 9 Morph-detection performance when trained/tested on J2 images.

	Train	Test	D-EER (in %)	∅	∅
J2	FaceFusion	FaceMorpher	2.26	7.34	8.32
		OpenCV	17.3		
		UBO-Morpher	14.63		
	FaceMorpher	FaceFusion	17.66	21.87	
		OpenCV	30.18		
		UBO-Morpher	27.26		
	OpenCV	FaceFusion	15.25	16.99	
		FaceMorpher	20.23		
		UBO-Morpher	18.99		
	UBO-Morpher	FaceFusion	12.32	10.06	
		FaceMorpher	4.88		
		OpenCV	20.23		
	ALL	*FaceFusion*	*11.29*	11.20	-
		FaceMorpher	*5.9*		
		OpenCV	*17.4*		
		UBO-Morpher	*14.94*		

Table 10 Morph-detection performance when trained/tested on PS-J2 images.

	Train	Test	D-EER (in %)	∅	∅
PS-J2	FaceFusion	FaceMorpher	2.93	9.65	8.52
		OpenCV	29.88		
		UBO-Morpher	17.3		
	FaceMorpher	FaceFusion	37.22	27.36	
		OpenCV	29.26		
		UBO-Morpher	24.69		
	OpenCV	FaceFusion	23.36	10.68	
		FaceMorpher	7.24		
		UBO-Morpher	12.53		
	UBO-Morpher	FaceFusion	25.82	10.27	
		FaceMorpher	3.85		
		OpenCV	16.89		
	ALL	*FaceFusion*	*24.49*	13.30	-
		FaceMorpher	*5.18*		
		OpenCV	*16.17*		
		UBO-Morpher	*14.12*		

elimination of image capture format-dependent information, so that morphing-specific information is again more likely to be considered during training. This explains the good average performance of 2.26%.

Again, the KDE plot in Fig. 10 clearly shows that training on Facemorpher shows the least competitive results. The corresponding red curve lies to a large extent to the left of the EER line. As also the corresponding bona fide curve is shifted to the left, the D-EER of 7.60% is still okay when training is done on FaceMorpher.

However, the step-like appearance of the MB-LBP plot shown in Fig. 8b and the straight sections on both sides of the curve indicate that the statistical significance in this case is limited, which is due to the size of the database used for testing in connection with the very good morph attack detection performance: Since the selection used for testing from the FRGCv2 database contains only 2,405 images, the number of incorrectly classified images is very low overall due to the very good morph attack detection performance, meaning that even a few misclassifications can have a large impact on the resulting D-EER. In future work it could therefore be investigated whether the achieved performance determined here can also be verified when testing is done with significantly larger databases.

5.1.4 JPEG2000: The detection performance rates are shown in Table 9. When compressing the images using the JPEG2000 method, so much information is lost that the effect from the two previous scenarios does not occur. With an average of 8.32%, the D-EER is almost 70% higher compared to the resized scenario. Also, the performance is not as robust as in the other scenarios with values ranging between 7.34% (FaceFusion) and 21.87% (FaceMorpher).

Fig. 11 shows that, as in the previous scenarios, the red curve indicating FaceMorpher lies far to the left of the EER line. In this case, it lies even further to the left than the thick green curve, which represents the average performance, explaining the high D-EER of 21.87% when training is done on FaceMorpher. However, as Fig. 8c clearly points out, the MB-LBP algorithm still performs significantly better than all state-of-the-art algorithms compared.

5.1.5 Print/Scan - JPEG2000: The detection performance rates are shown in Table 10. Compressing the printed and scanned images using the JPEG2000 algorithm dramatically deteriorates

morph attack detection performance up to 8.52%. However, it can also be see that the performance is not reduced compared to the JPEG2000 scenario, as the D-EER increases by only 0.2% percentage points. Also the DET plot (Fig. 8d) and the KDE plot (Fig. 12) of the two scenarios look very similar. This indicates that it is in fact only the JPEG2000 compression that affects performance and printing and subsequent scanning does not further degrade the morph attack performance.

5.1.6 Generalization: If looking at how the detection performance changes when different post-processings are used for training and evaluation, the trend already observed becomes apparent again. As one can see in Table 11 PS images are always detected the best. When evaluating J2 images the performance deteriorates significantly. While J2 images are only poorly detected, training on them provides relatively robust results. It can be assumed that depending on the application scenario, training with different post-processings is preferable. Future research might consider the potential effect on the robustness of the attack detection performance when a fusion of multiple post-processing is performed.

Table 11 Average detection performance (D-EER in %) for different post-processing.

	Test			
Train	RS	J2	PS	PS-J2
RS	5.80	14.53	3.03	27.72
J2	8.21	8.32	7.60	16.07
PS	15.45	16.99	2.26	14.01
PS-J2	12.32	12.68	4.26	8.52

5.2 Morph Attack Detection combined with Face Recognition System

It is worth investigating to what extent the detection performance of the proposed system is influenced by the use of a FRS. Using a COTS FRS, the threshold, which decides whether an image is accepted or rejected, is selected in such a way that a FMR of 0.1% is achieved. The proposed morphing detection system is applied for each reference face image which has been part of a biometric match of the FRS. If the FRS rejects many of the supposedly easier to detect morphs, these attacks fall out of the set of attacks that the morphing attack detection algorithm has to detect, which could lead to a relative deterioration of the detection performance. On the other hand, the bona fide images, which are incorrectly rejected by the FRS, also fall out of the set of images that the morphing attack detection algorithm has to examine. For this experiment the morphing attack detection system operates at the threshold of the D-EER point.

As can be seen in Fig. 13 the FRS incorrectly accepts over 90% of the attacks of which 77% are rejected by the morphing attack detection resulting in a APCER of 23%. It might also be of interest to note that nearly 20% of the morphs rejected by the FRS would have been accepted by the morphing attack detection, indicating that some of the morphs which are more easily detected by the FRS are potentially somewhat more difficult for the proposed system to detect. This shows the potential that lies in a combination of the two systems. However 100% of the bona fide images are recognized as such by the FRS, while the BPCER of the morphing attack detection system lies at 23%. Therefore APCER and BPCER are still identical and the average detection performance of the proposed system did not deteriorate despite the preceding use of the FRS.

6 Conclusion

In this paper, the performance of MB-LBP on morphs created by four different morphing algorithms is evaluated. In addition, the images were post-processed in various ways. The performance of single algorithms highly depends on the morphs used for training and testing. The robustness of the algorithms over different morphing

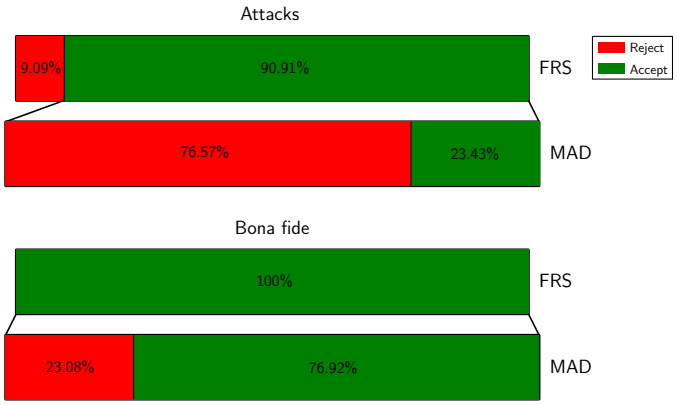


Fig. 13: Combination of FRS and morphing attack detection (MAD)

algorithms and databases can be increased by the fusion of multiple MB-LBP configurations and different cell divisions. Further, it is demonstrated that the robustness increases with the number of fused algorithms. Training on multiple attack types leads to a more robust morph detection performance and, therefore, lower error rates in some cases. The proposed MB-LBP fusion approach outperforms most of the state-of-the-art no-reference morph detection algorithms. Finally, this paper emphasizes the need for robust morphing detection algorithms and diverse databases comprised of different image sources and morphing algorithms in order to reliably train and evaluate face morphing attack detection algorithms.

Acknowledgment

This research work has been funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE as well as by the Federal Office of Information Security (BSI) within the FACETRUST project. The UBO-Morpher was kindly provided by the University of Bologna.

7 References

1 J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2Face: Real-time Face Capture and Reenactment of RGB Videos," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.
2 J. Thies, M. Zollhöfer, and M. Niessner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 66:1–66:12, Jul. 2019. [Online]. Available: http://doi.acm.org/10.1145/3306346.3323035
3 A. Bharati, R. Singh, M. Vatsa, and K. W. Bowyer, "Detecting facial retouching using supervised deep learning," *IEEE Trans. on Information Forensics and Security*, vol. 11, no. 9, pp. 1903–1913, 2016.
4 C. Rathgeb, A. Dantcheva, and C. Busch, "Impact and detection of facial beautification in face recognition: An overview," *IEEE Access*, 2019.
5 U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face recognition systems under morphing attacks: A survey," *IEEE Access*, vol. 7, pp. 23 012–23 026, 2019.
6 J. T. A. Andrews, T. Tanay, and L. D. Griffin, "Multiple-identity image attacks against face-based identity verification."
7 N. Damer, A. M. Saladie, A. Braun, and A. Kuijper, "MorGAN: Recognition vulnerability and attack detectability of face morphing attacks created by generative adversarial network," in *Proceedings of the 9th IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE, 2018.
8 R. Raghavendra, K. B. Raja, and C. Busch, "Detecting morphed face images," in *Proceedings of the 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, sep 2016.
9 U. Scherhag, R. Raghavendra, K. B. Raja, M. Gomez-Barrero, C. Rathgeb, and C. Busch, "On the vulnerability of face recognition systems towards morphed face attacks," in *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Apr. 2017.
10 R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch, "Face morphing versus face averaging: Vulnerability and detection," in *Proceedings of the 2017 International Joint Conference on Biometrics (IJCB)*. IEEE, oct 2017.
11 R. Raghavendra, S. Venkatesh, K. Raja, and C. Busch, "Towards making morphing attack detection robust using hybrid scale-space colour texture features," in

Proceedings of 5th International Conference on Identity, Security and Behaviour Analysis (ISBA 2019), 2019, pp. 22–24.
12 ——, "Detecting face morphing attacks with collaborative representation of steerable features," in *Proceedings of the 3rd Computer Vision and Image Processing (CVIP2018)*, 2018, pp. 1–11.
13 A. Agarwal, R. Singh, M. Vatsa, and A. Noore, "SWAPPED! digital face presentation attack detection via weighted local magnitude pattern," in *Proceedings of the 2017 International Joint Conference on Biometrics (IJCB)*. IEEE, oct 2017.
14 A. Asaad and S. Jassim, "Topological data analysis for image tampering detection," in *Proceedings of the 10th International Workshop on Digital Forensics and Watermarking (IWDW)*. Springer International Publishing, 2017, pp. 136–146.
15 S. Jassim and A. Asaad, "Automatic detection of image morphing by topology-based analysis," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, 2018.
16 A. Makrushin, T. Neubert, and J. Dittmann, "Automatic generation and detection of visually faultless facial morphs," in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2017.
17 U. Scherhag, C. Rathgeb, and C. Busch, "Towards detection of morphed face images in electronic travel documents," in *Proceedings of the 13th IAPR Workshop on Document Analysis Systems (DAS)*, 2018.
18 ——, "Morph detection from single face images: a multi-algorithm fusion approach," in *Proceedings of the 2018 International Conference on Biometrics Engineering and Application (ICBEA)*. ACM, 2018.
19 ——, "Performance variation of morphed face image detection algorithms across different datasets," in *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Jun. 2018.
20 L. Spreeuwers, M. Schils, and R. Veldhuis, "Towards robust evaluation of face morphing detection," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, 2018.
21 L. Wandzik, G. Kaeding, and R. V. Garcia, "Morphing detection using a general-purpose face recognition system," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, 2018.
22 C. Kraetzer, A. Makrushin, T. Neubert, M. Hildebrandt, and J. Dittmann, "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 - IHMMSec '17*. ACM Press, 2017.
23 L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch, "PRNU-based detection of morphed face images," in *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2018.
24 L. Debiasi, C. Rathgeb, U. Scherhag, A. Uhl, and C. Busch, "PRNU variance analysis for morphed face image detection," in *Proceedings of the 9th IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE, 2018.
25 U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl, "Detection of face morphing attacks based on PRNU analysis," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, pp. 1–1, 2019.
26 L. Debiasi, N. Damer, A. M. Saladié, C. Rathgeb, U. Scherhag, C. Busch, F. Kirchbuchner, and A. Uhl, "On the detection of gan-based face morphs using established morph detectors," in *Proceedings of the 20th International Conference on Image Analysis and Processing (ICIAP)*, 2019.
27 L.-B. Zhang, F. Peng, and M. Long, "Face morphing detection using fourier spectrum of sensor pattern noise," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, jul 2018.
28 M. Hildebrandt, T. Neubert, A. Makrushin, and J. Dittmann, "Benchmarking face morphing forgery detection: Application of StirTrace for impact simulation of different processing steps," in *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, apr 2017.
29 C. Seibold, A. Hilsmann, and P. Eisert, "Reflection analysis for face morphing attack detection," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, 2018.
30 P. Korshunov and S. Marcel, "Vulnerability of face recognition to deep morphing."
31 R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch, "Transferable deep-CNN features for detecting digital and print-scanned morphed face images," in *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jul 2017.
32 C. Seibold, W. Samek, A. Hilsmann, and P. Eisert, "Detection of face morphing attacks by deep learning," in *Digital Forensics and Watermarking*. Springer International Publishing, 2017, pp. 107–120.
33 ——, "Accurate and robust neural networks for security related applications exampled by face morphing attacks," *Computer Vision and Pattern Recognition*, pp. 1–16, 2018.
34 N. Damer, S. Zienert, Y. Wainakh, A. M. Saladié, F. Kirchbuchner, and A. Kuijper, "A multi-detector solution towards an accurate and generalized detection of face morphing attacks," in *Proceedings of the 22nd International Conference on Information Fusion (FUSION)*, 2019.
35 N. Damer, A. M. Saladié, S. Zienert, Y. Wainakh, P. Terhörst, F. Kirchbuchner, and A. Kuijper, "To detect or not to detect: The right faces to morph," in *Proceedings of the 12th IAPR International Conference On Biometrics (ICB)*, 2019.
36 M. Ferrara, A. Franco, and D. Maltoni, "Face morphing detection in the presence of printing/scanning and heterogeneous image sources," *CoRR*, vol. abs/1901.08811, 2019. [Online]. Available: http://arxiv.org/abs/1901.08811
37 ——, "The magic passport," in *Proceedings of the 2014 International Joint Conference on Biometrics (IJCB)*. IEEE, sep 2014.
38 T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, jan 1996.
39 J. Kannala and E. Rahtu, "BSIF: Binarized statistical image features," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 1363–1366.
40 M. Ngan, P. Grother, K. Hanaoka, and J. Kuo, "Face Recognition Vendor Test (FRVT) Part 4: MORPH Performance of Automated Face Morph Detection," National Institute of Technology (NIST), Tech. Rep. NISTIR 8292, 2020.
41 P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.
42 P. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, apr 1998.
43 International Civil Aviation Organization, "ICAO doc 9303, machine readable travel documents – part 9: Deployment of biometric identification and electronic storage of data in MRTDs (7th edition)," ICAO, Tech. Rep., 2015.
44 M. Ferrara, A. Franco, and D. Maltoni, "Face demorphing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 1008–1017, apr 2018.
45 E. Commision, "Eu-passport-specification," European Commision, Tech. Rep., 2018.
46 U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R. N. J. Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel, R. Breithaupt, R. Raghavendra, and C. Busch, "Biometric systems under morphing attacks: Assessment of morphing techniques and vulnerability reporting," in *Proceedings of the 2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, sep 2017.
47 S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Advances in Biometrics*. Springer Berlin Heidelberg, 2007, pp. 828–837.
48 ISO/IEC JTC1 SC37 Biometrics, "Information technology – biometric presentation attack detection – part 3: Testing and reporting," International Organization for Standardization, Geneva, Switzerland, ISO ISO/IEC IS 30107-3:2017, 2017.
49 J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CoRR*, vol. abs/1801.07698, 2018. [Online]. Available: http://arxiv.org/abs/1801.07698