On the Effectiveness of Local Binary Patterns in Face Anti-spoofing

Ivana Chingovska, André Anjos, Sébastien Marcel

Idiap Research Institute
Centre du Parc - rue Marconi 19, CH-1920 Martigny, Suisse
{ivana.chingovska, andre.anjos, sebastien.marcel}@idiap.ch

Abstract: Spoofing attacks are one of the security traits that biometric recognition systems are proven to be vulnerable to. When spoofed, a biometric recognition system is bypassed by presenting a copy of the biometric evidence of a valid user. Among all biometric modalities, spoofing a face recognition system is particularly easy to perform: all that is needed is a simple photograph of the user.

In this paper, we address the problem of detecting face spoofing attacks. In particular, we inspect the potential of texture features based on Local Binary Patterns (LBP) and their variations on three types of attacks: printed photographs, and photos and videos displayed on electronic screens of different sizes. For this purpose, we introduce REPLAY-ATTACK, a novel publicly available face spoofing database which contains all the mentioned types of attacks. We conclude that LBP, with ∼15% Half Total Error Rate, show moderate discriminability when confronted with a wide set of attack types.

1 Introduction

Spoofing attack is the action of outwitting a biometric sensor by presenting a counterfeit biometric evidence of a valid user [NAR08]. It is a direct attack to the sensory input of a biometric system and the attacker does not need previous knowledge about the recognition algorithm. Most of the biometric modalities are not resistant to spoofing attacks: the biometric systems are usually designed to only recognize identities without concern whether the identity is live or not. Despite the existence of very sophisticated biometric authentication and verification systems nowadays, implementing anti-spoofing schemes for them is still in its infancy.

Depending on the biometric modality being attacked, fabricating fake biometric data can have different levels of difficulty. While creating an artificial finger to spoof a fingerprint recognition system, or printing contact lens to spoof an iris recognition system may require some expertise, it is very easy to create a copy of someone's face. All that is needed is a photograph of the person, which can be easily found on the Internet or taken directly from the user at distance. The assumptions that the artificial biometric evidence can bypass a biometric recognition system, are not only chimerical: in [DM09] the authors have shown how to successfully spoof a laptop authentication system using only a printed photograph.

Since face spoofing attracted the attention of the biometric community, a number of publications which address the problem in different ways have appeared [NAR08]. Possible options include engaging additional devices to detect if there is a live person in front of the camera, or asking the user to respond to some challenge, like making a particular gesture. However, completely automatic systems which do not rely on additional hardware and are not intrusive are cheaper and more convenient for the user.

The development of new algorithms to solve the problem is not possible without a public database which catalogs many different types of attacks. The purpose of this paper is two-fold. Firstly, we introduce a novel publicly available database, called REPLAY-ATTACK, with three types of attacks and accompanied by a protocol and a baseline study of its effectiveness in bypassing biometric recognition systems. Secondly, we study the strength of texture features based on Local Binary Patters (LBP) to discriminate between real access and a spoof attack. In support of the idea for reproducible research, the database is freely available for public use, while the method is simple and easy to re-implement. Its source code is also freely available for reproduction of results.

In what follows, we give a brief overview of the state-of-the-art anti-spoofing measures and the efforts in creating face spoofing database up to now in Section 2. In Section 3 we present REPLAY-ATTACK and its companion protocols. Section 4 describes the studied counter-measure, followed by experimental results in Section 5. Conclusions and directions for future research are given in Section 6.

2 Related work

Summary of existing anti-spoofing methods. The existing anti-spoofing methods generally move towards one of three directions: analyzing the texture of the image captured by the sensor, detecting any evidence of liveness on the scene or combining both approaches together. Focusing on the texture based methods, the first attempt towards spoofing detection was made in [L$^+$04], where the authors argue that the frequency distributions on the image of a live person and the image of an attack are different. In [B$^+$10], the authors decompose the face image into a specular (reflectance) and diffusion component. They conclude that recaptured images show reflectance characteristics of paper and ink and achieve Equal Error Rate (EER) of 6.7%. Using the Lambertian reflectance model and Difference of Gaussians (DoG) filtering, [T$^+$10] extracts two types of latent samples which are representatives of the texture of the image. The algorithms achieves an Area Under the ROC Curve (AUC) value of 0.95. The DoG approach is also employed in [Z$^+$12], where the image is processed with 4 DoG filters with different values of σ and the achieved EER is 17%.

Most recently, the approach proposed in [MHP11] exploits yet another texture feature, LBP. Each image is represented with a feature vector which is a concatenation of a $\text{LBP}^{u2}_{16,2}$ histogram over the whole image, $\text{LBP}^{u2}_{8,1}$ histograms of 9 overlapping blocks in which the image is divided, and a $\text{LBP}^{u2}_{8,1}$ histogram over the whole image. In the notation $\text{LBP}^{u2}_{P,R}$, the superscript stands for uniform LBP, while the subscripts refer to the number of points P which form the LBP code and are taken on a circle of radius R around the central pixel. The resulting multi-scale LBP based feature vectors have dimensionality of 833 and are fed to an SVM for a final classification.

Another category of anti-spoofing methods focus on detection of a live-face specific motion on the scene, such as eye-blinking, mouth movements or head movements. Examples of methods using eye-blinking detection are proposed in [PWS08] and [JUY06]. There are a number of publications which analyze specific properties of the human head as a 3D object and its movements, like [KFB09] or [B$^+$09]. Both methods use optical flow field for motion estimation and report EER of 0.5% and HTER of 10% respectively. [AM11] states that in the case of an attack using a photograph, there should be high correlation between the total amount of movement in the face region and the scene background. The algorithm achieves HTER of 8.98%.

Most of the mentioned papers test their methods on attacks with printed photographs. Both motion-based and texture-based anti-spoofing measures have confirmed their discriminability in such cases. In this work we go one step further: we exploit the capabilities of texture-based features to reveal the difference between real accesses and attacks with photographs and videos.

Summary of existing face spoofing databases. Many of the previously mentioned papers test their proposed counter-measures on databases which they have developed, but are not publicly available. The lack of publicly available databases and protocols obstructs fair evaluation and comparison of the different anti-spoofing methods. The first database designed specifically for development of anti-spoofing algorithms, is the NUAA Photograph Imposter Database (from this point onwards referred as NUAA) which accompanies [T$^+$10] and consists only of attacks with printed photographs. Its main disadvantage, besides the limited number of identities (15 in total), is the provision of still images instead of videos, which makes it unusable for motion-based algorithms. The protocol provided by NUAA can not be considered as complete, because it contains only training and test data and overlooks development data for fine tuning of the classifiers. In [Z$^+$12] the authors propose a face anti-spoofing database (from this point onward referred as CASIA-FASD) with three types of attacks: warped printed photographs, printed photographs with perforated eye regions and a video playbacks. In a certain sense, CASIA-FASD can be considered as an addendum of NUAA database, solving its two main drawbacks and adding attacks with video playbacks. However, it inherits the lack of a complete protocol. PRINT-ATTACK [AM11] is the first database which provides precise protocol with training, development and test set. It contains videos of attacks only with printed photographs to 50 different identities.

A database adequate for developing anti-spoofing algorithms should provide attacks capable of penetrating unprotected face recognition systems. Each database should provide an evaluation of the scores that a baseline face recognition system generates for spoof attacks to one identity. So far, NUAA and CASIA-FASD databases do not provide such evidence.

In Section 3 we present REPLAY-ATTACK, a novel face-spoofing database targeting to challenge the most advanced spoofing counter-measures. Not only it enriches PRINT-ATTACK by adding more diverse spoofing attacks, but it also provides a protocol for fair counter-measure comparison and proves the vulnerability of a face recognition system to its attacks.

3 The REPLAY-ATTACK database

The REPLAY-ATTACK biometric (face) database[1] consists of short video recordings of both real-access and attack attempts to 50 different identities. In this section we give the setup for database recording, describe the set of companion protocols and evaluate the effectiveness of the collected attacks.

Setup for database recording. To create the dataset each person recorded a number of videos at 2 different stationary conditions: (1) *controlled* (the background of the scene is uniform and the light of a fluorescent lamp illuminates the scene); and (2) *adverse* (the background of the scene is non-uniform and day-light illuminates the scene). People were asked to sit down in front of a custom acquisition system built on an Apple 13-inch MacBook laptop and capture two video sequences with a resolution of 320 by 240 pixels (QVGA), at 25 frames-per-second and of 15 seconds each (375 frames). The acquisition process is the same as for the PRINT-ATTACK database and it is thoroughly described in [AM11].

Collecting samples and generating the attacks. Under the same illumination and background settings used for real-access video clips, the acquisition operator took two high-resolution pictures of each person using a 12.1 megapixel Canon PowerShot SX150 IS camera and with an iPhone 3GS (3.1 megapixel camera), that would be used as basis for the spoofing attempts. To realize the attacks, the operator forges an attack as described in one of the following scenarios: (1) *print* (the operator displays hard copies of the high-resolution digital photographs printed on plain A4 paper using a Triumph-Adler DCC 2520 color laser printer); (2) *mobile* (the operator displays photos and videos taken with the iPhone using the iPhone screen); and (3) *highdef* (the operator displays the high-resolution digital photos and videos using an iPad screen with resolution (1024 by 768 pixels). Each attack video is captured for about 10 seconds in two different attack modes: (1) *hand-based attacks* (the operator holds the attack media or device using their own hands); and (2) *fixed-support attacks* (the operator sets the attack device on a fixed support so they do not move during the spoof attempt). The first set of (hand-based) attacks show a shaking behavior which can sometimes trick eye-blinking detectors [PWS08]. Figure 1 shows some frames of the captured spoofing attempts.

Protocols. The total set of videos in the database is decomposed into 3 subsets allowing for training, development and testing of binary classifiers. Identities for each subset were chosen randomly, but do not overlap, i.e. people that are on one of the subsets do not appear in any other set. Moreover, each attack subset can be sub-classified into two groups that split the attacking support used during the acquisition (hand-based or fixed-support). Counter-measures developed using this database can report error figures that consider both separated and aggregated grouping, from which it is possible to understand which types of attacks are better handled by the proposed method. Table 1 summarizes the number of videos taken for both real-access and attack attempts and how they are split in the different subsets and groups.

In the case the developed counter-measure requires training, it is recommended that training and development samples are used to train classifiers. One trivial example is to use

[1] `http://www.idiap.ch/dataset/replayattack`

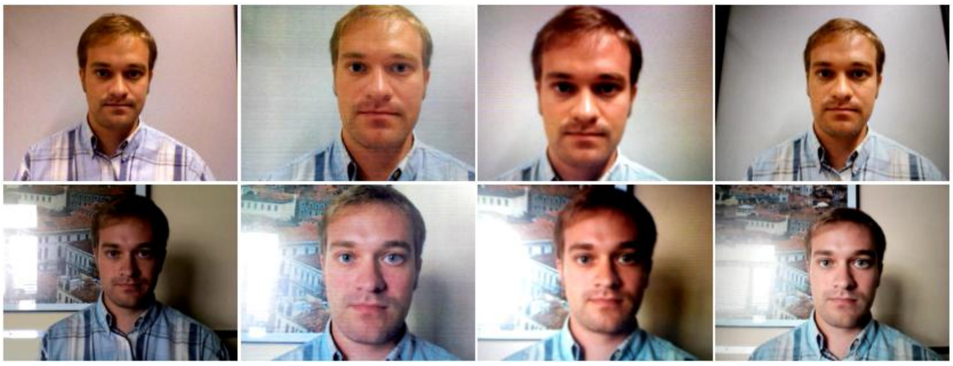

Figure 1: Examples of real accesses and attacks in different scenarios. In the top row, samples from *controlled* scenario. In the bottom row, samples from *adverse* scenario. Columns from left to right show examples of real access, printed photograph, mobile phone and tablet attacks.

Table 1: Number of videos in each database subset. Numbers displayed as sums indicate the amount of hand-based and fixed-support attacks available in each subset when relevant.

Type	Train	Devel.	Test	Total
Real-access	60	60	80	200
Print-attack	30+30	30+30	40+40	100+100
Phone-Attack	60+60	60+60	80+80	200+200
Tablet-Attack	60+60	60+60	80+80	200+200
Total	360	360	480	1200

the training set for training the classifier itself and the development data to estimate when to stop training. The test set should be **solely** used to report error rates and performance curves.

Effectiveness of Attacks. The REPLAY-ATTACK database provides an extra set of 100 videos that are not part of the "spoofing" scenarios defined above. The videos correspond to enrollment sequences for each of the 50 clients, in the two illumination conditions as described above. The enrollment videos are grouped respecting the same definitions for the data in spoofing attacks - identities do not overlap between training, development and test subsets. The videos should be used to train a baseline face-recognition classifier which can then be used to estimate the quality of attacks contained in the database.

4 Anatomy of the studied counter-measure

Simple visual inspection of an image of a real user and a recaptured image of the same user shows that the two images can be very similar and even the human eye may find it difficult to make a distinction at first glance (see Figure 1). Yet, some disparities between the real face and spoof-attack images may become evident once the images are translated into a proper feature space. These differences come from the fact that the human face as a 3D object, as well as the human skin, have their own optical qualities (absorption, reflection, scattering, refraction), which other materials (paper, photographic paper or electronic

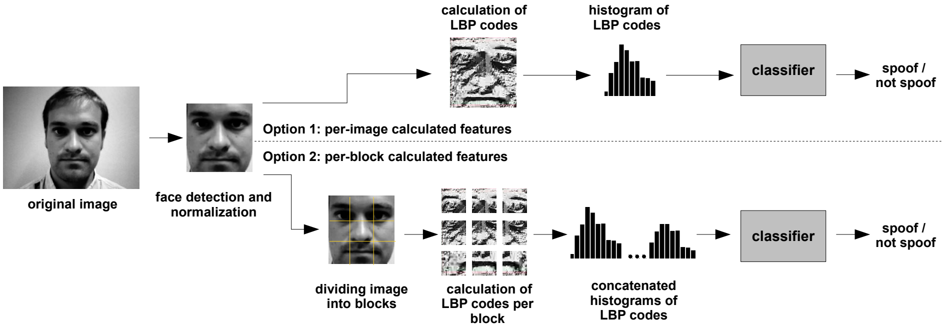
calculation of LBP codes
histogram of LBP codes
classifier
spoof / not spoof
original image
face detection and normalization
Option 1: per-image calculated features
Option 2: per-block calculated features
dividing image into blocks
calculation of LBP codes per block
concatenated histograms of LBP codes
classifier
spoof / not spoof

Figure 2: Block diagram of the proposed LBP based anti-spoofing algorithm

display) do not possess [PDT05]. Therefore, it is reasonable to assume that the texture properties of real accesses and spoof-attacks will be different.

In this work, we try to capture the texture properties of the images with features based on the Local Binary Patterns (LBP) operator introduced in [OPM02]. The most simple LBP pattern for a particular pixel, usually denoted as $\text{LBP}_{3\times3}$, is formed by comparing the intensity values of that pixel with the intensity values of the pixels in its 3x3 neighborhood. In this way, each pixel is assigned a label with value from 0 to $2^8 - 1$. In the case of uniform LBP (LBP^{u2}), only the labels which contain at most two 0-1 or 1-0 transitions are considered. The feature vector of an image, or a region of the image, is formed by calculating a histogram of the pixel labels.

Figure 2 displays a flow diagram for a complete overview of the counter-measure. The feature vectors are computed per frame. In this way, each frame of the videos in REPLAY-ATTACK and CASIA-FASD databases is considered as an independent sample in both training and testing sets. This choice enables fair comparison with the NUAA database which only provides images. The computed feature vectors apply to face bounding boxes normalized to 64x64 pixels. Unfortunately, the face detection process is not error free. In case a certain frame in the input video stream presents no detected face, the face detection is borrowed from any previous frame which had one. If there is no previous frame with detected face, the frame is discarded from further analysis.

The feature vector which is used for spoofing detection in this work is a simple normalized histogram of $\text{LBP}^{u2}_{3\times3}$ codes, as opposed to the concatenation of more complex parametrized LBP proposed in [MHP11]. As depicted in Figure 2, we calculate the LBP histogram in two different ways, and perform all the experiments separately on the both versions of the feature vectors. The first option is to calculate the LBP features for all pixels in the image and distribute them in one histogram (per-image calculated features). In this case, the total number of bins in the histogram, and thus the number of dimensions of the feature vector is 59. The second option is to divide the image into 3x3 blocks, calculate the LBP histograms for each of the blocks separately and form the final feature vector by their concatenation (per-block computed features). This results in a feature vector with 531 dimensions. The motivation for using blocks comes from the fact that the texture artifacts of the spoof attacks may be more visible in small and local uniform areas of the image, such as the forehead or the cheeks [MHP11].

Our work also includes experimentation with alternative LBP features, namely, the set of extended LBP as proposed in [TM10]. It includes transitional (tLBP), direction-coded (dLBP) and modified LBP (mLBP). The tLBP operator forms the binary patterns by comparing two consecutive neighboring pixels of the central pixel circularly in clockwise direction. The dLBP encodes the intensity variation along the four base directions through the central pixel in two bits. The mLBP, similarly to Modified Census Transform (MCT) features proposed in [FE04], compares the values of the neighboring pixels to the average of the intensity values in the 3x3 neighborhood. Unlike MCT, it discards the comparison of the central pixel with the average from the final code.

Since the feature vectors that we obtain from the images are histograms, the first classifier applied to the extracted features is χ^2 histogram comparison. We create reference histograms for the real accesses by averaging the histograms of the corresponding training samples. Reference histogram of attacks is not considered because the attacks are of many different types and hence their mean histogram is not a good overall representative of the cluster of attacks. Then, the feature vectors of the probe images are assigned a score as a result of χ^2 comparison with the reference histogram. A similar approach has already shown very good performance on the PRINT-ATTACK database [C$^+$11].

More complex classifiers were examined as well: a linear one, Linear Discriminant Analysis (LDA) and a non-linear one, Support Vector Machine (SVM) with radial kernel basis function.

The studied algorithm is implemented using the free signal-processing and machine learning toolbox Bob[2] [A$^+$12] and the source code of the algorithm is available as an anti-spoofing satellite package[3].

5 Experiments and results

In this section we give a performance evaluation of the studied anti-spoofing algorithm. Before proceeding with the experiments, we give an overview of the methodology we use to report the results and we report on the effectiveness of the attacks in REPLAY-ATTACK to spoof a baseline face recognition system.

5.1 Performance measure

A spoofing detection system is subject to two types of errors, either the real access is rejected (false rejection) or an attack is accepted (false acceptance). Its performance is often measured with Half Total Error Rate (HTER), which is half of the sum of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR). Since both the FAR and the FRR depend on a threshold τ, increasing the FAR will usually reduce the FRR and vice-versa. For this reason, results are often presented using the Receiver Operating Characteristic (ROC) curve, which plots the FAR versus the FRR for different values of τ. Another widely used measure is the Equal Error Rate (EER), defined as the point along the ROC curve where the FAR equals the FRR. The threshold τ should be chosen on the development set and the HTER reported using the test set. As means of uniforming reports, we

[2] `http://www.idiap.ch/software/bob/`

[3] `https://github.com/bioidiap/antispoofing.lbp`

recommend choosing the threshold τ on the EER at the development set.

5.2 Measuring the effectiveness of the attacks

To demonstrate the effectiveness of the REPLAY-ATTACK spoofs to get through a face recognition system, we implemented a Parts-Based Gaussian Mixture Model (PB-GMM) baseline[4] [CSM03]. According to a recent experimental evaluation of face recognition methods [W$^+$12], this method provides a trade-off in terms of complexity and performance accuracy. For this system, if one sets a threshold on the EER of the development set, the number of attacks that would be incorrectly classified as clients would be slightly above 82%. This validates the attacks in REPLAY-ATTACK as valuable for further investigation of counter-measures. As no enrollment data is provided for NUAA and CASIA-FASD databases, it is not possible to do such an evaluation for these databases.

5.3 Performance evaluation of the studied anti-spoofing algorithm

We provide a number of performance evaluation tables which give the results of the classification with respect to different criteria. Firstly, we compare the effectiveness of the different types of LBP to detect spoofing attacks. Secondly, taking only the best-performing type of LBP from the first experiment, we compare the performance of features computed per-image and per-block. Our third experiment evaluates the classification methods. It also compares the algorithm's performance to the reimplementation of the algorithm proposed in [MHP11]. Finally, we apply the studied counter-measure and the algorithm given in [MHP11] to all the available databases.

Comparison of the types of LBP. Inspired by the performance that the χ^2 statistics based classifier has shown on attacks with printed photographs [C$^+$11], we made an evaluation of regular LBP$^{u2}_{3\times3}$ and the extended set of LBP [TM10] using this classification method on REPLAY-ATTACK. The HTER of the classification for both the development and test set for per-image computed LBP histograms is given in Table 2.

Table 2: HTER (%) of classification with χ^2 for different types of LBP features

LBP$^{u2}_{3\times3}$		**tLBP**		**dLBP**		**mLBP**	
dev	test	dev	test	dev	test	dev	test
31.24	**34.01**	29.37	35.35	36.71	40.26	32.29	33.68

Although very efficient for printed spoofing attacks, probably due to apparent printing artifacts, LBP is not as discriminative with the other types of attacks in REPLAY-ATTACK. As it can be seen from Table 2, the lowest HTER on the test set is achieved by using mLBP, followed by LBP$^{u2}_{3\times3}$. Despite the very tight superiority of mLBP over LBP$^{u2}_{3\times3}$, we will continue the further evaluation only using LBP$^{u2}_{3\times3}$. The motivation for this lies in the fact that LBP$^{u2}_{3\times3}$ is much more commonly used then mLBP and this makes the comparison of the algorithms more consistent. Furthermore, computing mLBP requires one additional operation for calculating the average of the intensity values in the 3x3 neighborhood of a given pixel.

Per-image vs. per-block calculated features. Our second experiment compares the performance of χ^2 classifier with respect to the features computed per-image and per-

[4] Code available at: `https://github.com/bioidiap/antispoofing.verification.gmm`

block. Their dimensions are 59 and 531 respectively. The results are given in Table 3. We can conclude that enlarging the feature vector 9 times does not improve the performance, as the HTER in the first case is about the same as in the second case on both development and test set. Therefore, in the rest of the experiments with REPLAY-ATTACK we will consider only the features computed on the full image, which is more optimal in terms of computational and memory resources.

Table 3: HTER (%) of classification with χ^2 for LBP codes computed per-image and per-block

per-image computed LBP$^{u2}_{3\times3}$		**per-block computed LBP$^{u2}_{3\times3}$**	
dev	test	dev	test
31.24	**34.01**	33.66	34.30

Comparison of the classifiers. In the following experiment, we compare the simple χ^2 statistics method to more sophisticated classifiers (LDA and SVM). We also reimplemented the algorithm proposed in [MHP11], where the feature vector with length of 833 is composed of concatenated LBP histograms and the used classifier is also a SVM. The results are given in Table 4 .

Table 4: HTER (%) of classification with different classifiers

LBP$^{u2}_{3\times3}$ + χ^2		**LBP$^{u2}_{3\times3}$ + LDA**		**LBP$^{u2}_{3\times3}$ + SVM**		**LBP [MHP11] + SVM**	
dev	test	dev	test	dev	test	dev	test
31.24	34.01	19.60	17.17	14.84	15.16	13.90	13.87

While the improvement of the results when using LDA over the classification with χ^2 is notable, introducing non-linearity with SVM into the classification system decreases the HTER on the test set by only 2%. This comes with the cost of a very large SVM with a lot of support vectors, which means low computational performance during the classification. For example, the total number of support vectors is 25717 for feature vectors of length 59. Features proposed in [MHP11] bring additional improvement of 1.3%, but they are 14 times bigger then our features and the total size of the SVM is even larger in this case.

Performance of the algorithm on different databases. Finally, in Table 5 we present the performance of the algorithm on all the available face-spoofing databases. As an addition, we report the results that we obtained with the reimplemented version of the method proposed in [MHP11].

It should be remarked that, due to the lack of development set on NUAA and CASIA-FASD databases, the only option was to evaluate the algorithm using cross-validation by randomly dividing the training data into 5 folds. The results presented for these databases are actually the average HTER on the test set over 5 iterations of the algorithm with different folds playing the role of a development set.

It is also interesting to point out that when experimenting with NUAA and CASIA-FASD, better results were achieved with per-block computed features. Therefore, in Table 5, the features for REPLAY-ATTACK are computed per-image, while the features for the other two databases are computed per-block.

From Table 5, we discuss three crucial matters. Firstly, we observe the generalization capabilities of the algorithm on the NUAA database. Secondly, we examine the gain in

Table 5: HTER (%) of the classification on different databases. The average standard deviation of the HTER over the 5 cross-fold validation iterations on NUAA and CASIA-FASD is at most 1%.
∗Features replicated from [MHP11] using Bob and classification performed using Bob.
‡Result reported in [MHP11]: features generated using the Matlab implementation of LBP and classification performed using LIBSVM.
§Features from [MHP11] provided by the authors and classification performed using Bob.

	REPLAY-ATTACK		**NUAA**		**CASIA-FASD**	
	dev	test	dev	test	dev	test
$\mathbf{LBP}^{u2}_{3\times3}$ + LDA	19.60	17.17	0.06	18.32	17.08	21.01
$\mathbf{LBP}^{u2}_{3\times3}$ + SVM	14.84	15.16	0.11	19.03	16.00	18.17
LBP [MHP11] + SVM ∗	13.90	13.87	0.11	13.17	15.43	18.21
LBP [MHP11] + SVM ‡	-	-	-	2.5	-	-
LBP [MHP11] + SVM §	-	-	3.21	4.23	-	-

precision versus the complexity of the classifier. Lastly, we highlight a disparity between the results obtained with our reimplementation of the method in [MHP11] and the results reported by the authors.

The first thing to notice in Table 5 is the tendency of the classifiers to overfit on the training set of the NUAA database. Both LDA and SVM yield high performance on the development set, but are less effective on the test set. This can be explained by the fact that the classification threshold is chosen on the development set, which for NUAA is actually a subset of the training set, as we perform cross-validation. This problem can be taken as an indication for the necessity of a precise protocol with separate training, development and test set in spoof-attack databases.

From Table 5 we can also observe that the non-linear classifier gives just minor improvement over LDA not only in REPLAY-ATTACK, as shown before, but also on NUAA and CASIA databases. Moreover, expanding the feature vector to very high dimensionality like in [MHP11] does not guarantee good generalization for NUAA, nor does help for better performance on CASIA database.

Table 5 contains two additional rows. The first one is the HTER on the NUAA database as reported in [MHP11]. We can observe that there is a difference between the HTER we obtained by reimplementing the method in [MHP11] (13.17%) and the reported value (2.5%). Therefore, we asked the authors to send us, for comparison purposes, their features generated using the Matlab implementation of LBP[5]. The last row of Table 5 gives the HTER of the algorithm that we obtained with the features they provided to us, and it amounts 4.23%. The small difference between this value and the value reported in the paper is probably due to a different cross-validation schemes in the both settings. However, the question remains what causes the disparity between the results obtained with the two implementations.

After thorough investigation, we found that the Matlab implementation of LBP responds unexpectedly in certain conditions. On the contrary, the LBP implementation of Bob ap-

[5] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

pears to handle these conditions correctly. Namely, when calculating the circular LBP$^{u2}_{8,1}$ and LBP$^{u2}_{16,2}$ required for the features in [MHP11], there is a need for calculating interpolated values of particular points in the image. The Matlab implementation of LBP does not handle correctly bilinear interpolation in some cases due to precision problems, leading to completely different LBP codes. The anomaly seems to affect $\sim$4% of the LBP codes for some of the test images. We informed the authors about this issue for further investigation, especially to understand why the anomaly produces better performance.

6 Conclusion

Spoofing and anti-spoofing has become a prevalent topic in the biometrics community. Regardless of the sophistication of a particular face recognition system, it should not be completely trusted if it does not have a protection against spoofing attacks.

The contributions of this paper can be summarized as follows. Firstly, it introduces REPLAY-ATTACK, a novel spoofing attack database containing three types of possible attacks using three different media and two different recording conditions. The database includes a protocol for training, development and testing purposes, and also proves the vulnerability of a baseline face recognition system to its attacks. Secondly, it proposes simple and easily reproducible LBP based face spoofing counter-measure and explores its efficiency against a variety of attacks. Variants of LBP were also investigated, but the regular LBP$^{u2}_{3\times3}$ shows the best performance/complexity tradeoff. The simple setup and low-dimensional features manage to reach reasonable performance even without using complex non-linear classifiers whose size can be inconvenient for fast computation. In support of reproducible research, the database, its protocols, as well as the source code will be made publicly available.

The LBP based anti-spoofing method guarantees different levels of certainty for different types of attacks and different databases. Some attacks can deceive this counterfeit more easily then others. There is no consistency in the results with regards to the types of attacks, nor the attacks from different databases. Our belief is that this is not valid only for texture-based methods, but also for methods that approach the problem from different aspect. The various face spoofing attacks differ from the real accesses in their own particular manner: the devices that are used introduce different artifacts and the amount and type of movement they possess is different. In other words, the cues that distinguish two different types of face spoofing attacks from real accesses differ in their essence and should be grasped in their own unique way. There is not a single notion which describes all the types of attacks. Hence, we believe that the future work in the field of anti-spoofing should focus on addressing as more spoofing attacks as possible with separate attack-specific approaches. Another option is to congregate the characteristics of all the real accesses into a single model to which none of the spoofing attacks will relate in any sense.

Acknowledgments

The authors would like to thank the FP7 European TABULA RASA Project (257289) for its financial support, as well as Jukka Määttä, Abdenour Hadid and Matti Pietikäinen from University of Oulu, Finland, for the fruitful collaboration.

References

[A$^+$12] A. Anjos et al. Bob: a free signal processing and machine learning toolbox for researchers. In *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*. ACM Press, October 2012.

[AM11] A. Anjos and S. Marcel. Counter-Measures to Photo Attacks in Face Recognition: a public database and a baseline. In *International Joint Conference on Biometrics 2011*, 2011.

[B$^+$09] W. Bao et al. A Liveness Detection Method for Face Recognition Based on Optical Flow Field. *2009 International Conference on Image Analysis and Signal Processing*, 2009.

[B$^+$10] J. Bai et al. Is Physics-based Liveness Detection Truly Possible with a Single Image? In *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2010.

[C$^+$11] M. M. Chakka et al. Competition on Counter Measures to 2-D Facial Attacks. In *International Joint Conference on Biometrics 2011*, 2011.

[CSM03] F. Cardinaux, C. Sanderson, and S. Marcel. Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS. In *Proceedings of the 4th International Conference on AVBPA*, University of Surrey, Guildford, UK, 2003.

[DM09] N. M. Duc and B. Q. Minh. Your face is not your password. Black Hat Conference, 2009.

[FE04] B. Froba and A. Ernst. Face Detection with the Modified Census Transform. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[JUY06] H. K. Jee, S. U.Jung, and J. H. Yoo. Liveness Detection for Embedded Face Recognition System. *Engineering and Technology*, 2006.

[KFB09] K. Kollreider, H. Fronthaler, and J. Bigun. Non-intrusive liveness detection by face images. *Image and Vision Computing*, 27(3), 2009.

[L$^+$04] J. Li et al. Live Face Detection Based on the Analysis of Fourier Spectra. *Biometric Tecnology for Human Identification*, 2004.

[MHP11] J. Määttä, A. Hadid, and M. Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *Proc. International Joint Conference on Biometrics (IJCB 2011), Washington, D.C., USA*, 2011.

[NAR08] K.A. Nixon, V. Aimale, and R. K. Rowe. Spoof Detection Schemes. *Handbook of Biometrics*, 2008.

[OPM02] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7), 2002.

[PDT05] G. Parziale, J. Dittman, and M. Tistarelli. Analysis and evaluation of alternatives and advanced solutions for system elements. BioSecure D 9.1.2, 2005.

[PWS08] G. Pan, Z. Wu, and L. Sun. Liveness Detection for Face Recognition. *Recent Advances in Face Recognition*, December 2008.

[T$^+$10] X. Tan et al. Face Liveness Detection from a Single Image with Sparse Low Rank Bilinear Discriminative Model. In *ECCV (6)*, 2010.

[TM10] J. Trefny and J. Matas. Extended Set of Local Binary Patterns for Rapid Object Detection. Computer Vision Winter Workshop, Czech Republic, 2010.

[W$^+$12] R. Wallace et al. Cross-pollination of normalisation techniques from speaker to face authentication using gaussian mixture models. *IEEE Transactions on Information Forensics and Security*, 2012.

[Z$^+$12] Z. Zhiwei et al. A Face Antispoofing Database with Diverse Attacks. In *Proceedings of the 5th IAPR International Conference on Biometrics (ICB'12), New Delhi, India*, 2012.