Spoofing Estimation using Deep Convolutions Networks for Iris, Face And Fingerprint

G. Jaya Lakshmi[1], B. Bhaskar Reddy[2]
[1]PG Scholar, Dept of ECE, Bheema Institute of Technology & Science, AP, India,
[2]Associate Professor, Dept of ECE, Bheema Institute of Technology & Science, AP, India

Abstract: In the existing system, the iris anti-spoofing methods have explored hardcoded features through image-quality metrics, texture patterns, bags-of-visual words and noise artifacts due to the recapturing process. The performance of such solutions vary significantly from dataset to dataset. In the case of face anti-spoofing method, the available solutions in the literature mostly deal with the face spoofing detection problem through texture patterns (e.g., LBP-like detectors), acquisition telltales (noise), and image quality metrics. In this method we approach the problem by extracting meaningful features directly from the data regard- less of the input. This project focuses on the automatically extract vision meaningful features directly from the data using deep representation. Assuming a very limited knowledge about biometric spoofing at the sensor to derive outstanding spoofing detection systems for iris, face and fingerprint modalities based on two deep learning approaches. The first approach consists of learning suitable convolutional network architectures for each domain, whereas the second approach focuses on learning the weights of the network via back propagation. We consider nine biometric spoofing benchmarks—each one containing real and fake samples of a given biometric modality and attack type— and learn deep representations for each benchmark by combining and contrasting the two learning approaches. The results strongly indicate that spoofing detection systems based on convolution networks can be robust to attacks already known and possibly adapted, with little effort, to image-based attacks that are yet to come.

Keywords: Deep Learning, Convolutional Networks, Hyper Parameter Architecture Optimization, Filter Weights Learning, Back-Propagation, Spoofing Detection.

I. INTRODUCTION

Biometrics human characteristics and traits can successfully allow people identification and authentication and have been widely used for access control, surveillance, and also in national and global security systems [1]. In the last few years, due to the recent technological improvements for data acquisition, storage and processing, and also the scientific advances in computer vision, pattern recognition, and machine learning, several biometric modalities have been largely applied to person recognition, ranging from traditional fingerprint to face, to iris, and, more recently, to vein and blood flow. Simultaneously, various spoofing attacks techniques have been created to defeat such biometric systems. There are several ways to spoof a biometric system [2], [3]. Indeed, previous studies show at least eight different points of attack [4], [5] that can be divided into two main groups: direct and indirect attacks. The former considers the possibility to generate synthetic biometric samples, and is the first vulnerability point of a biometric security system acting at the sensor level. The latter includes all the remaining seven points of attacks and requires different levels of knowledge about the system, e.g., the matching algorithm used, the specific feature extraction procedure, database access for manipulation, and also possible weak links in the communication channels within the system.

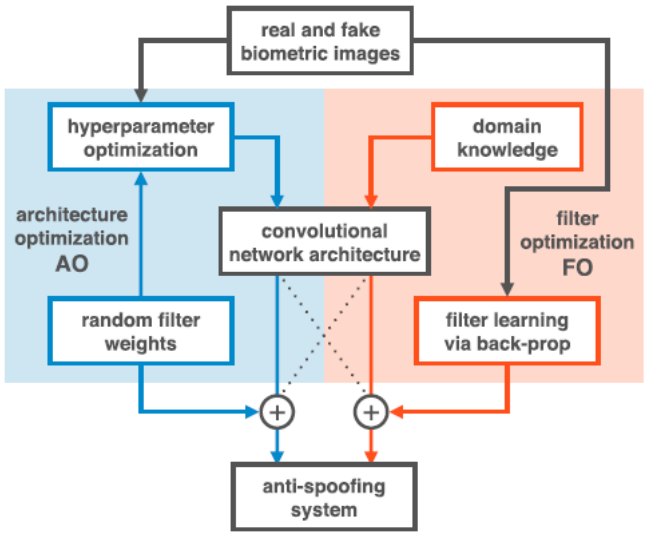


Fig. 1. Schematic diagram detailing how anti-spoofing systems are built from spoofing detection benchmarks. Architecture optimization (AO) is shown on the left and filter optimization (FO) on the right.

Given that the most vulnerable part of a system is its acquisition sensor, attackers have mainly focused on direct spoofing. This is possibly because a number of biometric traits can be easily forged with the use of common apparatus and consumer electronics to imitate real biometric readings (e.g., stampers, printers, displays, audio recorders). In response to that, several biometric spoofing benchmarks have been recently proposed, allowing researchers to make steady progress in the conception of anti-spoofing systems. Three relevant modalities in which spoofing detection has been investigated are iris, face, and fingerprint. Benchmarks across these modalities usually share the common characteristic of being image or video-based. In the context of irises, attacks are normally performed using

printed iris images [6] or, more interestingly, cosmetic contact lenses [7], [8]. With faces, impostors can present to the acquisition sensor a photography, a digital video [9], or even a 3D mask [10] of a valid user. For fingerprints, the most common spoofing method consists of using artificial replicas [11] created in a cooperative way, where a mold of

In this work, we not only evaluate AO and FO in separate, but also in combination, as indicated by the crossing dotted lines. The fingerprint is acquired with the cooperation of a valid user and is used to replicate the user's fingerprint with different materials, including gelatin, latex, play-doh or silicone. The success of an anti-spoofing method is usually connected to the modality for which it was designed. In fact, such systems often rely on expert knowledge to engineer features that are able to capture acquisition telltales left by specific types of attacks. However, the need of custom-tailored solutions for the myriad possible attacks might be a limiting constraint. Small changes in the attack could require the redesign of the entire system. In this paper, we do not focus on custom-tailored solutions. Instead, inspired by the recent success of Deep Learning in several vision tasks [12]–[16], and by the ability of the technique to leverage data, we focus on two general-purpose approaches to build image-based anti-spoofing systems with convolutional networks for several attack types in three biometric modalities, namely iris, face, and fingerprint. The first technique that we explore is hyper parameter optimization of network architectures [17], [18] that we henceforth call architecture optimization, while the second lies at the core of convolutional networks and consists of learning filter weights via the well-known back-propagation [19] algorithm, hereinafter referred to as filter optimization.

Fig. 1 illustrates how such techniques are used. The architecture optimization (AO) approach is presented on the left and is highlighted in blue while the filter optimization (FO) approach is presented on the right and is highlighted in red. As we can see, AO is used to search for good architectures of convolutional networks in a given spoofing detection problem and uses convolutional filters whose weights are set at random in order to make the optimization practical. This approach assumes little a priori knowledge about the problem, and is an area of research in deep learning that has been successful in showing that the architecture of convolutional networks, by themselves, is of extreme importance to performance [17], [18], [20]–[23]. In fact, the only knowledge AO assumes about the problem is that it is approachable from a computer vision perspective.

Still in Fig 1, FO is carried out with back-propagation in predefined network architecture. This is a longstanding approach for building convolutional networks that has recently enabled significant strides in computer vision, specially because of an understanding of the learning process, and the availability of plenty of data and processing power [13], [16], [24]. Network architecture in this context is usually determined by previous knowledge of related problems. In general, we expect AO to adapt the architecture to the problem in hand and FO to model important stimuli for discriminating fake and real biometric samples.

We evaluate AO and FO not only in separate, but also in combination, i.e., architectures learned with AO are used for FO as well as previously known good performing architectures are used with random filters. This explains the crossing dotted lines in the design flow of Fig 1. As our experiments show, the benefits of evaluating AO and FO apart and later combining them to build anti-spoofing systems are twofold. First, it enables us to have a better comprehension of the interplay between these approaches, something that has been largely underexplored in the literature of convolutional networks. Second, it allows us to build systems with outstanding performance in all nine publicly available benchmarks considered in this work. The first three of such benchmarks consist of spoofing attempts for iris recognition systems, Biosec [25], Warsaw [26], and MobBIOfake [27]. Replay-Attack [9] and 3DMAD [10] are the benchmarks considered for faces, while Biometrika, CrossMatch, Italdata, and Swipe are the fingerprint benchmarks here considered, all them recently used in the 2013 Fingerprint Liveness Detection Competition (LivDet'13) [11]. Results outperform state-of-the-art counterparts in eight of the nine cases and observe a balance in terms of performance between AO and FO, with one performing better than the other depending on the sample size and problem difficulty.

In some cases, we also show that when both approaches are combined, we can obtain performance levels that neither one can obtain by itself. Moreover, by observing the behaviour of AO and FO, we take advantage of domain knowledge to propose a single new convolutional architecture that push performance in five problems even further, sometimes by a large margin, as in CrossMatch (68.80% v. 98.23%). The experimental results strongly indicate that convolutional networks can be readily used for robust spoofing detection. Indeed, we believe that data-driven solutions based on deep representations might be a valuable direction to this field of research, allowing the construction of systems with little effort even to image-based attack types yet to come. We organized the remainder of this work into five sections. Section II presents previous anti-spoofing systems for the three biometric modalities covered in this paper, while Section III presents the considered benchmarks. Section IV describes the methodology adopted for architecture optimization (AO) and filter optimization (FO) while Section V presents experiments, results, and comparisons with state-of-the-art methods. Finally, Section VI concludes the paper and discusses some possible future directions.

II. RELATED WORK

In this section, we review anti-spoofing related work for iris, face, and fingerprints, our focus in this paper.

A. Iris Spoofing

Daugman [28, Sec. 8—Countermeasures against Subterfuge]1 was one of the first authors to discuss the feasibility of some attacks on iris recognition systems. The author proposed the use of Fast Fourier Transform to verify the high frequency spectral magnitude in the frequency domain. The solutions for iris liveness detection available in the literature range from active solutions relying on special acquisition hardware [30]–[32] to software-based solutions relying on texture analysis of the effects of an attacker using color contact lenses with someone

else's pattern printed onto them [33]. Software-based solutions have also explored the effects of cosmetic contact lenses [7], [8], [34], [35]; pupil constriction[36]; and multi biometrics of electroence-phalogram (EEG) and iris together [37], among others. Galbally et al. [38] investigated 22 image quality measures (e.g., focus, motion, occlusion, and pupil dilation). The best features are selected through sequential floating feature selection (SFFS) [39] to feed a quadratic discriminant classifier. The authors validated the work on the BioSec [25], [40] benchmark. Sequeira et al. [41] also explored image quality measures [38] and three classification techniques validating the work on the BioSec [25], [40] and Clarkson [42] benchmarks and introducing the MobBIOfake benchmark comprising 800 iris images from the MobBIO multimodal database [27].

Sequeira et al. [43] extended upon previous works also exploring quality measures. They first used a feature selection step on the features of the studied methods to obtain the "best features" and then used well-known classifiers for the decision making. In addition, they applied iris segmentation [44] to obtaining the iris contour and adapted the feature extraction processes to the resulting non-circular iris regions. The validation considered five datasets (BioSec [25], [40], MobBIOfake [27], Warsaw [26], Clarkson [42] and NotreDame [45]. Textures have also been explored for iris liveness detection. In the recent MobILive2 [6] iris spoofing detection competition, the winning team explored three texture descriptors: Local Phase Quantization (LPQ) [46], Binary Gabor Pattern [47], and Local Binary Pattern (LBP) [48]. Analyzing printing regularities left in printed irises, Czajka [26] explored some peaks in the frequency spectrum were associated to spoofing attacks. For validation, the authors introduced the Warsaw dataset containing 729 fake images and 1,274 images of real eyes. In [42], The First Intl. Iris Liveness Competition in 2013, the Warsaw database was also evaluated; however, the best reported result achieved 11.95% of FRR and 5.25% of FAR by the University of Porto team. Sun et al. [49] recently proposed a general framework for iris image classification based on a Hierarchical Visual Codebook (HVC). The HVC encodes the texture primitives of iris images and is based on two existing bag-of-words models. The method achieved state-of-the-art performance for iris spoofing detection, among other tasks.

In summary, iris anti-spoofing methods have explored hardcoded features through image-quality metrics, texture patterns, bags-of-visual-words and noise artifacts due to the recapturing process. The performance of such solutions vary significantly from dataset to dataset. Differently, here we propose the automatically extract vision meaningful features directly from the data using deep representations.

B. Face Spoofing

We can categorize the face anti-spoofing methods into four groups [50]: user behavior modeling, methods relying on extra devices [51], methods relying on user cooperation and, finally, data-driven characterization methods. In this section, we review data-driven characterization methods proposed in literature, the focus of our work herein. Määttä et al. [52] used LBP operator for capturing printing artifacts and micro-texture patterns added in the fake biometric samples during acquisition. Schwartz et al. [50] explored color, texture, and shape of the face region and used them with Partial Least Square (PLS) classifier for deciding whether a biometric sample is fake or not. Both works validated the methods with the Print Attack benchmark [53]. Lee et al. [54] also explored image-based attacks and proposed the frequency entropy analysis for spoofing detection. Pinto et al. [55] pioneered research on video-based face spoofing detection. They proposed visual rhythm analysis to capture temporal information on face spoofing attacks. Mask-based face spoofing attacks have also been considered thus far. Erdogmus et al. [56] dealt with the problem through Gabor wavelets: local Gabor binary pattern histogram sequences [57] and Gabor graphs [58] with a Gabor-phase based similarity measure [59]. Erdogmus & Marcel [10] introduced the 3D Mask Attack database (3DMAD), a public available 3D spoofing database, recorded with Microsoft Kinect sensor.

Kose et al. [60] demonstrated that a face verification system is vulnerable to mask-based attacks and, in another work, Kose et al. [61] evaluated the anti-spoofing method proposed by Määttä et al. [52] (originally proposed to detect photo based spoofing attacks). Inspired by the work of Tan et al. [62], Kose et al. [63] evaluated a solution based on reflectance to detect attacks performed with 3D masks. Finally, Pereira et al. [64] proposed a score-level fusion strategy in order to detect various types of attacks. In a follow up work, Pereira et al. [65] proposed an anti-spoofing solution based on the dynamic texture, a spatio-temporal version of the original LBP. Results showed that LBP-based dynamic texture description has a higher effectiveness than the original LBP. In summary, similarly to iris spoofing detection methods, the available solutions in the literature mostly deal with the face spoofing detection problem through texture patterns (e.g., LBP-like detectors), acquisition telltales (noise), and image quality metrics. Here, we approach the proplem by extracting meaningful features directly from the data regardless of the input type (image, video, or 3D masks).

TABLE I: Main Features of the Benchmarks Considered Herein

Modality	Benchmark/Dataset	Color	Dimension cols × rows	# Training			# Testing			# Development		
				Live	Fake	Total	Live	Fake	Total	Live	Fake	Total
Iris	Warsaw [26]	No	640 × 480	228	203	431	624	612	1236			
	Biosec [25]	No	640 × 480	200	200	400	800	800	1200			
	MobBIOfake [27]	Yes	250 × 200	400	400	800	400	400	800			
Face	Replay-Attack [77]	Yes	320 × 240	600	3000	3600	4000	800	4800	600	3000	3600
	3dMad [78]	Yes	640 × 480	350	350	700	250	250	500	250	250	500
Fingerprint	Biometrika [11]	No	312 × 372	1000	1000	2000	1000	1000	2000			
	CrossMatch [11]	No	800 × 750	1250	1000	2250	1250	1000	2250			
	Italdata [11]	No	640 × 480	1000	1000	2000	1200	1000	2000			
	Swipe [11]	No	208 × 1500	1221	979	2200	1153	1000	2153			

C. Fingerprint Spoofing

We can categorize fingerprint spoofing detection methods roughly into two groups: hardware-based (exploring extra sensors) and software-based solutions (relying only on the information acquired by the standard acquisition sensor of the authentication system) [11]. Galbally et al. [66] proposed a set of feature for fingerprint liveness detection based on quality measures such as ridge strength or directionality, ridge continuity, ridge clarity, and integrity of the ridge-valley structure. The validation considered the three benchmarks used in LivDet 2009 – Fingerprint competition [67] captured with

different optical sensors: Biometrika, CrossMatch, and Identix. Later work [68] explored the method in the presence of gummy fingers. Ghiani et al. [69] explored LPQ [46], a method for representing all spectrum characteristics in a compact feature representation form. The validation considered the four benchmarks used in the LivDet 2011 – Fingerprint competition [70]. Gragnaniello et al. [71] explored the Weber Local Image Descriptor (WLD) for liveness detection, well suited to high contrast patterns such as the ridges and valleys of fingerprints images. In addition, WLD is robust to noise and illumination changes. The validation considered the LivDet 2009 and LivDet 2011 – Fingerprint competition datasets. Jia et al. [72] proposed a liveness detection scheme based on Multi-scale Block Local Ternary Patterns (MBLTP).

Differently of the LBP, the Local Ternary Pattern operation is done on the average value of the block instead of the pixels being more robust to noise. The validation considered the LivDet 2011 – Fingerprint competition benchmarks. Ghiani et al. [73] explored Binarized Statistical Image Features (BSIF) originally proposed by Kannala et al. [74]. The BSIF was inspired in the LBP and LPQ methods. In contrast to LBP and LPQ approaches, BSIF learns a filter set by using statistics of natural images [75]. The validation considered the LivDet 2011 – Fingerprint competition benchmarks. Recent results reported in the LivDet 2013 Fingerprint Liveness Detection Competition [73] show that fingerprint spoofing attack detection task is still an open problem with results still far from a perfect classification rate. We notice that most of the groups approach the problem with hard-coded features sometimes exploring quality metrics related to the modality (e.g., directionality and ridge strength), general texture patterns (e.g., LBP-, MBLTP-, and LPQ-based methods), and filter learning through natural image statistics. This last approach seems to open a new research trend, which seeks to model the problem learning features directly from the data. We follow this approach in this work, assuming little a priori knowledge about acquisition-level biometric spoofing and exploring deep representations of the data.

D. Multi-Modalities

Recently, Galbally et al. [76] proposed a general approach based on 25 image quality features to detect spoofing attempts in face, iris, and fingerprint biometric systems. Our work is similar to theirs in goals, but radically different with respect to the methods. Instead of relying on prescribed image quality features, we build features that would be hardly thought by a human expert with AO and FO. Moreover, here we evaluate our systems in more recent and updated benchmarks.

III. BENCHMARKS

In this section, we describe the benchmarks (datasets) that we consider in this work. All of them are publicly available upon request and suitable for evaluating countermeasure methods to iris, face and fingerprint spoofing attacks. Table I shows the major features of each one and in the following we describe their details.

A. Iris Spoofing Benchmarks

Biosec: This benchmark was created using iris images from 50 users of the BioSec [25]. In total, there are 16 images for each user (2 sessions × 2 eyes × 4 images), totalizing 800 valid access images. To create spoofing attempts, the original images from Biosec were preprocessed to improve quality and printed using an HP Deskjet 970cxi and an HP LaserJet 4200L printers. Finally, the iris images were recaptured with the same iris camera used to capture the original images.

Warsaw: This benchmark contains 1274 images of 237 volunteers representing valid accesses and 729 printout images representing spoofing attempts, which were generated by using two printers: (1) a HP LaserJet 1320 used to produce 314 fake images with 600 dpi resolution, and (2) a Lexmark C534DN used to produce 415 fake images with 1200 dpi resolution. Both real and fake images were captured by an IrisGuard AD100 biometric device.

MobBIOfake: This benchmark contains live iris images and fake printed iris images captured with the same acquisition sensor, i.e., a mobile phone. To generate fake images, the authors first performed a preprocessing in the original images to enhance the contrast. The preprocessed images were then printed with a professional printer on high quality photographic paper.

B. Video-Based Face Spoofing Benchmarks

Replay-Attack: This benchmark contains short video recordings of both valid accesses and video-based attacks of 50 different subjects. To generate valid access videos, each person was recorded in two sessions in a controlled and in an adverse environment with a regular webcam. Then, spoofing attempts were generated using three techniques:

- Print attack, which presents to the acquisition sensor hard copies of high-resolution digital photographs printed with a Triumph-Adler DCC 2520 color laser printer;
- Mobile attack, which presents to the acquisition sensor photos and videos taken with an iPhone using the iPhone screen; and high-definition attack, in which high resolution photos and videos taken with an iPad are presented to the acquisition sensor using the iPad screen.
- **3DMAD:** This benchmark consists of real videos and fake videos made with people wearing masks. A total of 17 different subjects were recorded with a Microsoft Kinect sensor, and videos were collected in three sessions. For each session and each person, five videos of 10 seconds were captured. 3D masks were produced by ThatsMyFace.com using one frontal and two profile images of each subject.

All videos were recorded by the same acquisition sensor.

C. Fingerprint Spoofing Benchmarks

LivDet2013: This dataset contains four sets of real and fake fingerprint readings performed in four acquisition sensors: Biometrika FX2000, Italdata ET10, Crossmatch L Scan Guardian, and Swipe. For a more realistic scenario, fake samples in Biometrika and Italdata were generated without user

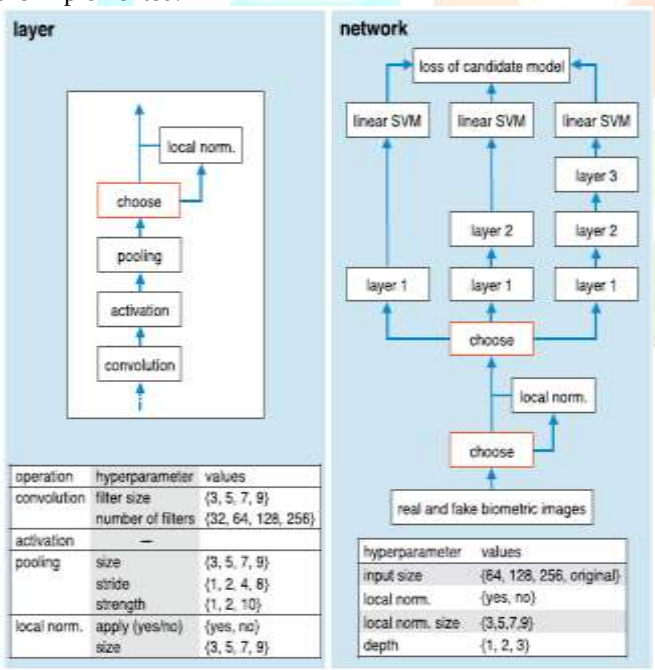
cooperation, while fake samples in Crossmatch and Swipe were generated with user cooperation. Several materials for creating the artificial fingerprints were used, including gelatin, silicone, latex, among others.

D. Remark

Images found in these benchmarks can be observed in Fig. 5 of Section V. As we can see, variability exists not only
across how operations are stacked in a layer (left) and how the network is instantiated and evaluated according to possible hyper parameter values (right). Note that a three-layered convolutional network of this type has a total of 25 hyper parameters governing both its architecture and its overall behavior through a particular instance of stacked operation modalities, but also within modalities. Moreover, it is rather unclear what features might discriminate real from spoofed images, which suggests that the use of a methodology able to use data to its maximum advantage might be a promising idea to tackle such set of problems in a principled way.

IV. METHODOLOGY

In this section, we present the methodology for architecture optimization (AO) and filter optimization (FO) as well as details about how benchmark images are preprocessed, how AO and FO are evaluated across the benchmarks, and how these methods are implemented.



operation	hyperparameter	values
convolution	filter size	{3, 5, 7, 9}
	number of filters	{32, 64, 128, 256}
activation	–	
pooling	size	{3, 5, 7, 9}
	stride	{1, 2, 4, 8}
	strength	{1, 2, 10}
local norm.	apply (yes/no)	{yes, no}
	size	{3, 5, 7, 9}

hyperparameter	values
input size	{64, 128, 256, original}
local norm.	{yes, no}
local norm. size	{3,5,7,9}
depth	{1, 2, 3}

Fig. 2. Schematic diagram for architecture optimization (AO) illustrating.

A. Architecture Optimization (AO)

Our approach for AO builds upon the work of Pinto et al. [17] and Bergstra et al. [23], i.e., fundamental, feedforward convolutional operations are stacked by means of hyper parameter optimization, leading to effective yet simple convolutional networks that do not require expensive filter optimization and from which prediction is done by linear support vector machines (SVMs). Operations in convolutional networks can be viewed as linear and non-linear transformations that, when stacked, extract high level representations of the input. Here we use a well-known set of operations called (i) convolution with a bank of filters, (ii) rectified linear activation, (iii) spatial pooling, and (iv) local normalization. Appendix provides a detailed definition of these operations. We denote as layer the combination of these four operations in the order that they appear in the left panel of Fig. 2. Local normalization is optional and its use is governed by an additional "yes/no" hyper parameter. In fact, there are other six hyper parameters, each of a particular operation, that have to be defined in order to instantiate a layer. They are presented in the lower part of the left panel in Fig. 2 and are in accordance to the definitions of Appendix. Considering one layer and possible values of each hyper parameter, there are over 3,000 possible layer architectures, and this number grows exponentially with the number of layers, which goes up to three in our case (Fig. 2 right panel).

In addition, there are network-level hyper parameters, such as the size of the input image, that expand possibilities to amyriad potential architectures. The overall set of possible hyper parameter values is called search space, which in this case is discrete and contains variables that are only meaningful in combination with others. For example, hyper parameters of a given layer are just meaningful if the candidate architecture has actually that number of layers. In spite of the intrinsic difficulty in optimizing architectures in this space, random search has played and important role in problems of this type [17], [18] and it is the strategy of our choice due to its effectiveness and simplicity. We can see in Fig. 2 that a three-layered network has a total of 25 hyper parameters, seven per layer and four at network level. They are all defined in Appendix with the exception of input size, which seeks to determine the best size of the image's greatest axis (rows or columns) while keeping its aspect ratio. Concretely, random search in this paper can be described as follows:

- Randomly — and uniformly, in our case — sample values from the hyper parameter search space;
- Extract features from real and fake training images with the candidate architecture;
- Evaluate the architecture according to an optimization objective based on linear SVM scores;
- Repeat steps 1–3 until a termination criterion is met;
- Return the best found convolutional architecture.

Even though there are billions of possible networks in the search space (Fig. 2), it is important to remark that not all candidate networks are valid. For example, a large number of candidate architectures (i.e., points in the search space) would produce representations with spatial resolution smaller than one pixel. Hence, they are naturally unfeasible. Additionally, in order to avoid very large representations, we discard in advance candidate architectures whose intermediate layers produce representations of over 600K elements or whose output representation has over 30K elements. Filter weights are randomly generated for AO. This strategy has been successfully used in the vision literature [17], [20], [21], [79] and is essential to make AO practical, avoiding the expensive filter optimization

(FO) part in the evaluation of candidate architectures. We sample weights from a uniform distribution U(0, 1) and normalize the filters to zero mean and unit norm in order to ensure that they are spread over the unit sphere. When coupled with rectified linear activation (Appendix), this sampling enforces sparsity in the network by discarding about 50% of the expected filter responses, thereby improving the overall robustness of the feature extraction. A candidate architecture is evaluated by first extracting deep representations from real and fake images and later training hard-margin linear SVMs (C=105) on these representations.

We observed that the sensitivity of the performance measure was saturating with traditional 10-fold cross validation (CV) in some benchmarks. Therefore, we opted for a different validation strategy. Instead of training on nine folds and validating on one, we train on one fold and validate on nine. Precisely, the optimization objective is the mean detection accuracy obtained from this adapted cross-validation scheme, which is maximized during the optimization. For generating the 10 folds, we took special care in putting all samples of an individual in the same fold to enforce robustness to cross-individual spoofing detection in the optimized architectures. Moreover, in benchmarks where we have more than one attack type (e.g., Replay-Attack and LivDet2013, see Section III), we evenly distributed samples of each attack type across all folds in order to enforce that candidate architectures are also robust to different types of attack. Finally, the termination criterion of our AO procedure simply consists of counting the number of valid candidate architectures and stopping the optimization when this number reaches 2,000.

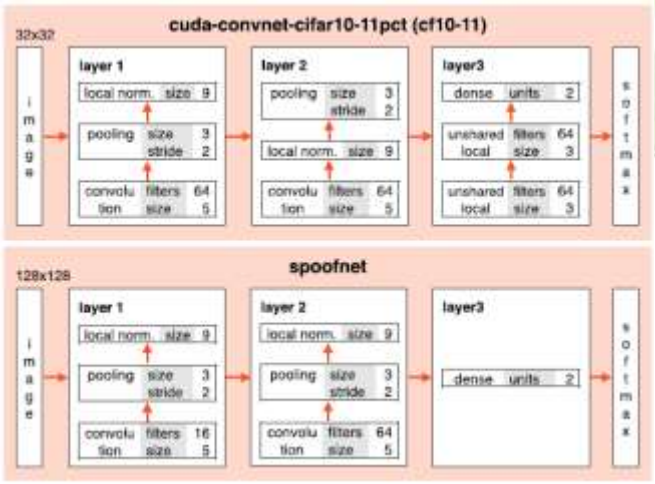


Fig. 3. Architecture of convolutional network found in the Cuda-convnet library [80] and here used as reference for filter optimization (cf10-11, top).

B. Filter Optimization (FO)

We now turn our attention to a different approach for tackling the problem. Instead of optimizing the architecture, we explore the filter weights and how to learn them for better characterizing real and fake samples. Our approach for FO is at the origins of convolutional networks and consists of learning filter weights via the well-known back-propagation algorithm [19]. Indeed, due to a refined understanding of the optimization process and the availability of plenty of data and processing power, back-propagation has been the gold standard method in deep networks for computer vision in the last years [13], [24], [81]. For optimizing filters, we need to have an already defined architecture. We start optimizing filters with a standard public convolutional network and training procedure. This network is available in the Cuda-convnet library [80] and is currently one of the best performing architectures in CIFAR-10,3 a popular computer vision benchmark in which such network achieves 11% of classification error. Hereinafter, we call this network cuda-convnet-cifar10-11pct, or simply cf10-11. Fig. 3 depicts the architecture of cf10-11 in the top panel and is a typical example where domain knowledge has been incorporated for increased performance. We can see it as a three-layered network in which the first two layers are convolutional, with operations similar to the operations used in architecture optimization (AO). In the third layer, cf10-11 has two sublayers of unshared local filtering and a final fully-connected sublayer on top of which softmax regression is performed. A detailed explanation of the operations in cf10-11 can be found in [80].

Proposed network architecture extending upon cf10-11 to better suiting spoofing detection problems (spoofnet, bottom). Both architectures are typical examples where domain knowledge has been incorporated for increased performance. In order to train cf10-11 in a given benchmark, we split the training images into four batches observing the same balance of real and fake images. After that, we follow a procedure similar to the original4 for training cf10-11 in all benchmarks, which can be described as follows:

- For 100 epochs, train the network with a learning rate of 10−3 by considering the first three batches for training and the fourth batch for validation;
- For another 40 epochs, resume training now considering all four batches for training;
- Reduce the learning rate by a factor of 10, and train the network for another 10 epochs;
- Reduce the learning rate by another factor of 10, and train the network for another 10 epochs.

After evaluating filter learning on the cf10-11 architecture, we also wondered how filter learning could benefit from an optimized architecture incorporating domain-knowledge of the problem. Therefore, extending upon the knowledge obtained with AO as well as with training cf10-11 in the benchmarks, we derived a new arc`hitecture for spoofing detection that we call spoofnet. Fig. 3 illustrates this architecture in the bottom panel and has three key differences as compared to cf10-11. First, it has 16 filters in the first layer instead of 64. Second, operations in the second layer are stacked in the same order that we used when optimizing architectures (AO). Third, we removed the two unshared local filtering operations in the third layer, as they seem inappropriate in a problem where object structure is irrelevant. These three modifications considerably dropped the number of weights in the network and this, in turn, allowed us to increase of size of the input images from 32×32 to 128×128. This is the fourth and last modification in spoofnet, and we

believe that it might enable the network to be more sensitive to subtle local patterns in the images.

TABLE II: Input Image Dimensionality After Basic Preprocessing On Face And Fingerprint Images (Highlighted) See Section Iv-C For Details

Modality	Benchmark	Dimensions *columns* × *rows*
Iris	Warsaw [26]	640 × 480
	Biosec [25]	640 × 480
	MobBIOfake [27]	250 × 200
Face	Replay-Attack [77]	**200 × 200**
	3DMAD [78]	**200 × 200**
Fingerprint	Biometrika [11]	**312 × 372**
	CrossMatch [11]	**480 × 675**
	Italdata [11]	**384 × 432**
	Swipe [11]	**187 × 962**

In order to train spoofnet, the same procedure used to train cf10-11 is considered except for the initial learning rate, which is made 10−4, and for the number of epochs in each step, which is doubled. These modifications were made because of the decreased learning capacity of the network. Finally, in order to reduce over fitting, data augmentation is used for training both networks according to the procedure of [13]. For cf10-11, five 24 × 24 image patches are cropped out from the 32×32 input images. These patches correspond to the four corners and central region of the original image, and their horizontal reflections are also considered. Therefore, ten training samples are generated from a single image. For spoofnet, the procedure is the same except for the fact that input images have 128 × 128 pixels and cropped regions are of 112×112 pixels. During prediction, just the central region of the test image is considered.

C. Elementary Preprocessing

A few basic preprocessing operations were executed on face and fingerprint images in order to properly learn representations for these benchmarks. This preprocessing led to images with sizes as presented in Table II and are described in the next two sections.

Face Images: Given that the face benchmarks considered in this work are video-based, we first evenly subsample 10 frames from each input video. Then, we detect the face position using Viola & Jones [82] and crop a region of 200 × 200 pixels centered at the detected window.

Fingerprint Images: Given the diverse nature of images captured from different sensors, here the preprocessing is defined according to the sensor type.

- **Biometrika:** We cropped the central region of size in columns and rows corresponding to 70% of the original image dimensions.
- **Italdata and CrossMatch:** We cropped the central region of size in columns and rows respectively corresponding to 60% and 90% of the original image columns and rows.
- **Swipe:** As the images acquired by this sensor contain a variable number of blank rows at the bottom, the average number of non-blank rows M was first calculated from the training images. Then, in order to obtain images of a common size with non-blank rows, we removed their blank rows at the bottom and rescaled them to M rows.

Finally, we cropped the central region corresponding to 90% of original image columns and M rows. The rationale for these operations is based on the observation that fingerprint images in LivDet2013 tend to have a large portion of background content and therefore we try to discard such information that could otherwise mislead the representation learning process. The percentage of cropped columns and rows differs among sensors because they capture images of different sizes with different amounts of background. For architecture optimization (AO), the decision to use image color information was made according to 10-fold validation (see Section IV-A), while for filter optimization (FO), color information was considered whenever available for a better approximation with the standard cf10-11 architecture. Finally, images were resized to 32 × 32 or 128 × 128 to be taken as input for the cf10-11 and spoofnet architectures, respectively.

D. Evaluation Protocol

For each benchmark, we learn deep representations from their training images according to the methodology described in Section IV-A for architecture optimization (AO) and in Section IV-B for filter optimization (FO). We follow the standard evaluation protocol of all benchmarks and evaluate the methods in terms of detection accuracy (ACC) and half total error rate (HTER), as these are the metrics used to assess progress in the set of benchmarks considered herein. Precisely, for a given benchmark and convolutional network already trained, results are obtained by:

- Retrieving prediction scores from the testing samples;
- Calculating a threshold τ above which samples are predicted as attacks;
- Computing ACC and/or HTER using τ and test predictions.

The way that τ is calculated differs depending on whether the benchmark has a development set or not (Table I). Both face benchmarks have such a set and, in this case, we simply obtain τ from the predictions of the samples in this set. Iris and fingerprint benchmarks have no such a set, therefore τ is calculated depending on whether the convolutional network was learned with AO or FO. In case of AO, we calculate τ by joining the predictions obtained from 10-fold validation (see Section IV-A) in a single set of positive and negative scores, and τ is computed as the point that lead to an equal error rate (EER) on the score distribution under consideration. In case of FO, scores are probabilities and we assume $\tau = 0.5$. ACC and HTER are then trivially computed with τ on the testing set. It is worth noting that the Warsaw iris benchmark provides a supplementary testing set that here we merge with the original testing set in order to replicate the protocol of [42]. Moreover, given face benchmarks are video-based and that in our methodology we treat them as images (Section IV-C), we perform a score-level fusion of the samples from the same video according to the max rule [83]. This fusion is done before calculating τ .

E. Implementation

Our implementation for architecture optimization (AO) is based on Hyperopt-convnet [84] which in turn is based on Theano [85]. LibSVM [86] is used for learning the linear classifiers via Scikit-learn.5 The code for feature extraction runs on GPUs due to Theano and the remaining part is multithreaded and runs on CPUs. We extended Hyperopt-convnet in order to consider the operations and hyper parameters as described in Appendix and Section IV-A and we will make the source code freely available in [87]. Running times are reported with this software stack and are computed in an Intel i7 @3.5GHz with a Tesla K40 that, on average, takes less than one day to optimize an architecture — i.e., to probe 2,000 candidate architectures — for a given benchmark. As for filter optimization (FO), Cuda-convnet [80] is used. This library has an extremely efficient implementation to train convolutional networks via back-propagation on NVIDIA GPUs. Moreover, it provides us with the cf10-11 convolutional architecture taken in this work as reference for FO.

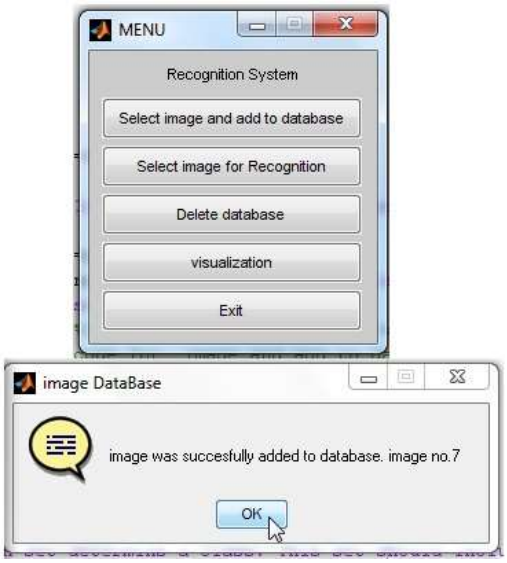


Fig6. Image added to database.

V. RESULT

Fig4. Menu

Fig7. Menu for Recognization

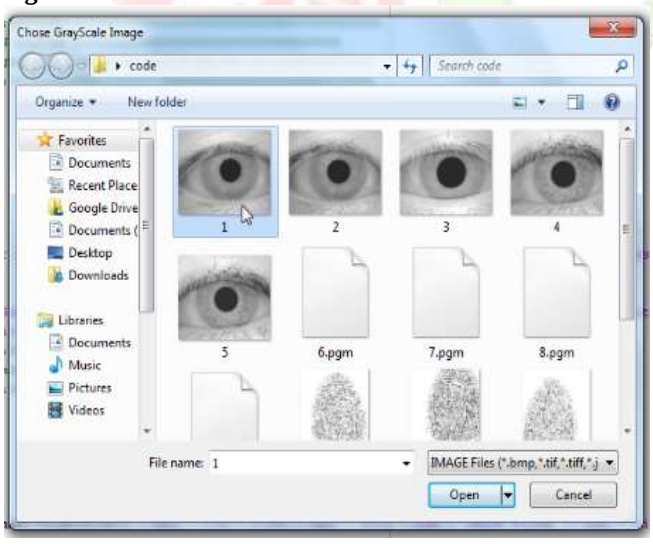


Fig5. Select Image

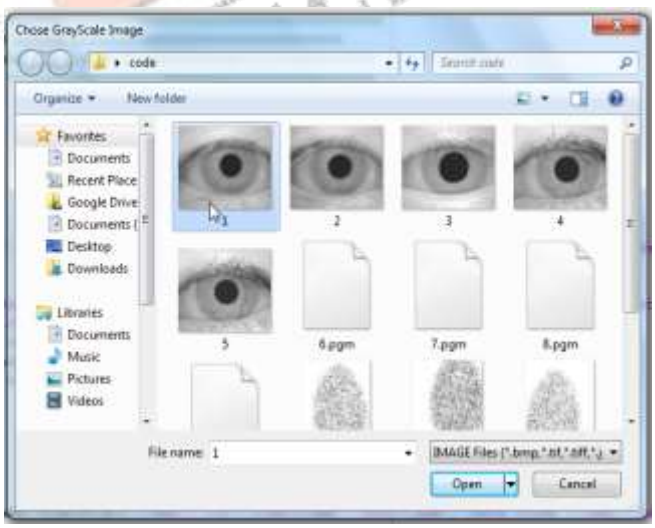


Fig8. Select Image for Reorganization

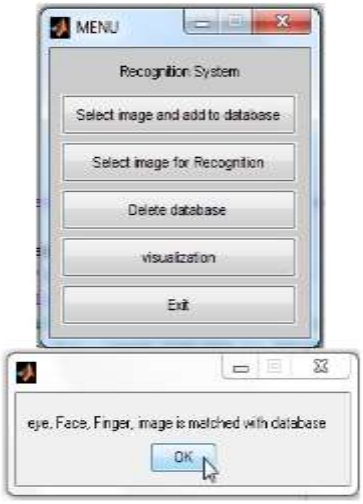


Fig9. Image Matched with Database.

Fig10. Select non-database Image

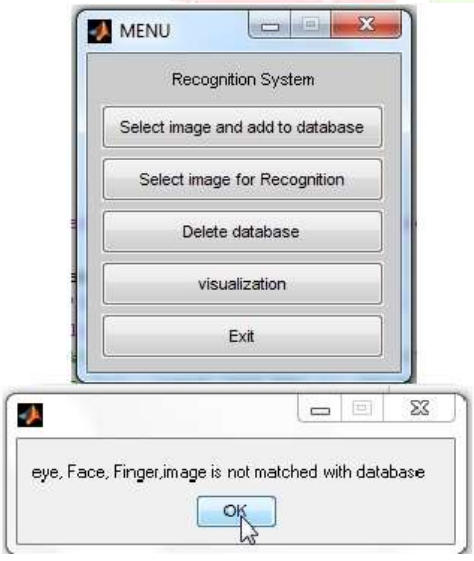


Fig11. Image Not matched with database.

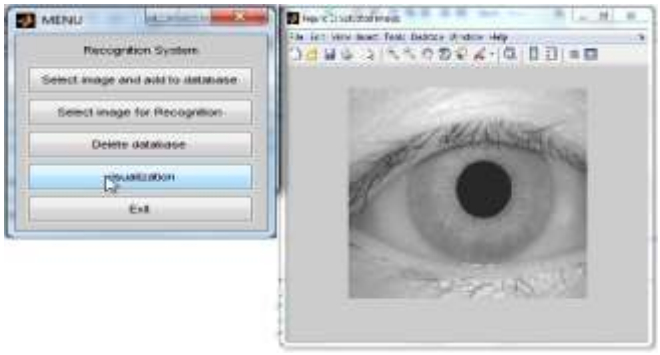


Fig12. Image Visualization

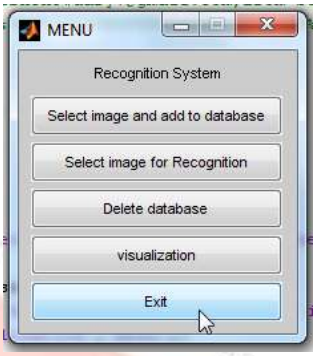


Fig13. Exit.

VI. CONCLUSION

In this project, investigated two deep representation research approaches for detecting spoofing in different biometric modalities. On one hand, approached the problem by learning representations directly from the data through architecture optimization with a final decision-making step atop the representations. On the other, to learn the filter weights for a given architecture using the well-known back- propagation algorithm. As the two approaches might seem naturally connected, we also examined their interplay when taken together. Experiments showed that these approaches achieved outstanding classification results for all problems and modalities outperforming the state-of-the-art results in eight out of nine benchmarks. Interestingly, the only case for which our approaches did not achieve SOTA results is for the biosec benchmark. These results support our hypothesis that the conception of data-driven systems using deep representations able to extract semantic and vision meaningful features directly from the data.

VI. REFERENCES

[1] A. K. Jain and A. Ross, "Introduction to biometrics," in Handbook of Biometrics. New York, NY, USA: Springer-Verlag, 2008, pp. 1–22.

[2] C. Rathgeb and A. Uhl, "Attacking iris recognition: An efficient hillclimbing technique," in Proc. IEEE/IAPR 20th Int. Conf. Pattern Recognit.(ICPR), Aug. 2010, pp. 1217–1220.

[3] C. Rathgeb and A. Uhl, "Statistical attack against iris-biometric fuzzy commitment schemes," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2011, pp. 23–30.

[4] J. Galbally, J. Fierrez, and J. Ortega-Garcia, "Vulnerabilities in biometric systems: Attacks and recent advances in liveness detection," Database, vol. 1, no. 3, pp. 1–8, 2007.
[5] N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in Audio- and Video-Based Biometric Person Authentication. Berlin, Germany: Springer-Verlag, 2001, pp. 223–228.
[6] A. F. Sequeira, H. P. Oliveira, J. C. Monteiro, J. P. Monteiro, and J. S. Cardoso, "MobILive 2014—Mobile iris liveness detection competition," in Proc. IEEE Int. Joint Conf. Biometrics (IJCB), Sep./Oct. 2014, pp. 1–6. [Online]. Available: http://mobilive2014. inescporto.pt/
[7] K. W. Bowyer and J. S. Doyle, "Cosmetic contact lenses and iris recognition spoofing," Computer, vol. 47, no. 5, pp. 96–98, May 2014.
[8] D. Yadav, N. Kohli, J. S. Doyle, R. Singh, M. Vatsa, and K. W. Bowyer, "Unraveling the effect of textured contact lenses on iris recognition," IEEE Trans. Inf. Forensics Security, vol. 9, no. 5, pp. 851–862, May 2014.
[9] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG), 2012, pp. 1–7.
[10] N. Erdogmus and S. Marcel, "Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect," in Proc. IEEE 6th Int. Conf.Biometrics, Theory, Appl., Syst. (BTAS), Sep./Oct. 2013, pp. 1–6.
[11] L. Ghiani et al., "LivDet 2013—Fingerprint liveness detection competition,"in Proc. Int. Conf. Biometrics (ICB), 2013, pp. 1–6. [Online].Available: http://prag.diee.unica.it/fldc/
[12] D. C. Cire¸san, U. Meier, L. M. Gambardella, and J. Schmidhuber,"Deep, big, simple neural nets for handwritten digit recognition," NeuralComput., vol. 22, no. 12, pp. 3207–3220, 2010.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classificationwith deep convolutional neural networks," in Advancesin Neural Information Processing Systems. Red Hook, NY, USA:Curran & Associates Inc., 2012.
[14] D. C. Cire¸san, U. Meier, J. Masci, and J. Schmidhuber, "Multi-columndeep neural network for traffic sign classification," Neural Netw., vol. 32,pp. 333–338, Aug. 2012.
[15] J. Ouyang and X. Wang, "Joint deep learning for pedestrian detection,"in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2014, pp. 2056–2063.
[16] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closingthe gap to human-level performance in face verification," in Proc. IEEEInt. Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1701–1708.
[17] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A high-throughputscreening approach to discovering good forms of biologically inspiredvisual representation," PLoS Comput. Biol., vol. 5, no. 11, p. e1000579,2009.
[18] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization,"J. Mach. Learn. Res., vol. 13, no. 1, pp. 281–305, 2012.
[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-basedlearning applied to document recognition," Proc. IEEE, vol. 86, no. 11,pp. 2278–2324, Nov. 1998.
[20] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng,"On random weights and unsupervised feature learning," in Proc. 28thInt. Conf. Mach. Learn., 2011, pp. 1–8.
[21] D. D. Cox and N. Pinto, "Beyond simple features: A large-scale featuresearch approach to unconstrained face recognition," in Proc. IEEE Int.Conf. Workshops Autom. Face Gesture Recognit. (FG), Mar. 2011,pp. 8–15.
[22] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithmsfor hyper-parameter optimization," in Advances in Neural InformationProcessing Systems. Red Hook, NY, USA: Curran & Associates Inc.,2011, pp. 2546–2554.
[23] J. S. Bergstra, D. Yamins, and D. D. Cox, "Making a science ofmodel search: **Hyper parameter** optimization in hundreds of dimensionsfor vision architectures," in Proc. 30th Int. Conf. Mach. Learn., 2013,pp. 115–123.
[24] K. Simonyan and A. Zisserman. (2014). "Very deep convolutionalnetworks for large-scale image recognition." [Online]. Available:http://arxiv.org/abs/1409.1556
[25] V. Ruiz-Albacete, P. Tome-Gonzalez, F. Alonso-Fernandez, J. Galbally,J. Fierrez, and J. Ortega-Garcia, "Direct attacks using fake imagesin iris verification," in Proc. 1st Eur. Workshop Biometrics IdentityManage. (BioID), vol. 5372. 2008, pp. 181–190.
[26] A. Czajka, "Database of iris printouts and its application: Developmentof liveness detection method for iris recognition," in Proc. 18th Int.Conf. Methods Models Autom. Robot. (MMAR), Aug. 2013, pp. 28–33.
[27] A. F. Sequeira, J. C. Murari, A. Rebelo, and H. P. Oliveira, "MobBIO:A multimodal database captured with a portable handheld device,"in Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP), 2014,pp. 133–139.
[28] J. Daugman, "Recognizing persons by their iris patterns," in Biometrics:Personal Identification in Networked Society. Boston, MA, USA:Kluwer, 1999, pp. 103–121.
[29] J. Daugman, "Iris recognition and anti-spoofing countermeasures," inProc. 7th Int. Biometrics Conf., 2004.
[30] E. C. Lee, K. R. Park, and J. Kim, "Fake iris detection by using purkinjeimage," in Advances in Biometrics (Lecture Notes in Computer Science),vol. 3832. New York, NY, USA: Springer-Verlag, 2005, pp. 397–403.
[31] A. Pacut and A. Czajka, "Aliveness detection for iris biometrics," inProc. 40th Annu. IEEE Int. Carnahan Conf. Secur. Technol., 2006,pp. 122–129.
[32] M. Kanematsu, H. Takano, and K. Nakamura, "Highly reliable livenessdetection method for iris recognition," in Proc. Annu. Conf. SICE,Sep. 2007, pp. 361–364.
[33] Z. Wei, X. Qiu, Z. Sun, and T. Tan, "Counterfeit iris detection basedon texture analysis," in Proc. 19th Int. Conf. Pattern Recognit. (ICPR),2008, pp. 1–4.
[34] N. Kohli, D. Yadav, M. Vatsa, and R. Singh, "Revisiting iris recognitionwith color cosmetic contact lenses," in Proc. IAPR Int. Conf.Biometrics (ICB), Jun. 2013, pp. 1–7.
[35] J. S. Doyle, K. W. Bowyer, and P. J. Flynn, "Variation in accuracyof textured contact lens detection based on sensor and lens pattern,"in Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl., Syst. (BTAS),Sep./Oct. 2013, pp. 1–7.
[36] X. Huang, C. Ti, Q.-Z. Hou, A. Tokuta, and R. Yang, "An experimentalstudy of pupil constriction for liveness detection,"

in Proc. IEEEWorkshop Appl. Comput. Vis. (WACV), Jan. 2013, pp. 252–258.
[37] T. Kathikeyan and B. Sabarigiri, "Countermeasures against IRIS spoofingand liveness detection using Electroencephalogram (EEG)," in Proc.Int. Conf. Comput., Commun., Appl. (ICCA), 2012, pp. 1–5.
[38] J. Galbally, J. Ortiz-Lopez, J. Fierrez, and J. Ortega-Garcia, "Iris livenessdetection based on quality related features," in Proc. IAPR Int. Conf.Biometrics (ICB), 2012, pp. 271–276.
[39] P. Pudil, J. Novovi˘cová, and J. Kittler, "Floating search methodsin feature selection," Pattern Recognit. Lett., vol. 15, no. 11,pp. 1119–1125, Nov. 1994.
[40] J. Fierrez-Aguilar, J. Ortega-Garcia, D. Torre-Toledano, andJ. Gonzalez-Rodriguez, "Biosec baseline corpus: A multimodalbiometric database," Pattern Recognit., vol. 40, no. 4, pp. 1389–1392,2007.
[41] A. F. Sequeira, J. Murari, and J. S. Cardoso, "Iris liveness detectionmethods in mobile applications," in Proc. Int. Conf. Comput. Vis. TheoryAppl. (VISAPP), 2014, pp. 22–33.
[42] S. Schuckers, K. W. Bowyer, A. Czajka, J. S. Doyle, and D. Yambay.(2013). LivDet 2013—Liveness Detection-Iris Competition. [Online].Available: http://people.clarkson.edu/projects/biosal/iris/
[43] A. F. Sequeira, J. Murari, and J. S. Cardoso, "Iris liveness detectionmethods in the mobile biometrics scenario," in Proc. Int. Joint Conf.Neural Netw. (IJCNN), 2014, pp. 3002–3008.
[44] J. C. Monteiro, A. F. Sequeira, H. P. Oliveira, and J. S. Cardoso, "Robustiris localisation in challenging scenarios," in Computer Vision, Imagingand Computer Graphics: Theory and Applications (Communications inComputer and Information Science). Berlin, Germany: Springer-Verlag,2004.
[45] J. Doyle and K. W. Bowyer. (2014). Notre Dame Image Database forContact Lens Detection in Iris Recognition—2013. [Online].
Available:http://www3.nd.edu/~cvrl/CVRL/Data_Sets.html, accessed Jun. 2014.
[46] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classificationusing local phase quantization," in Proc. 3rd Int. Conf. Image SignalProcess. (ICISP), 2008, pp. 236–243.
[47] Z. Zhang, Z. Zhou, and H. Li, "Binary Gabor pattern: An efficient androbust descriptor for texture classification," in Proc. 19th IEEE Int. Conf.Image Process. (ICIP), Sep./Oct. 2012, pp. 81–84.
[48] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scaleand rotation invariant texture classification with local binary patterns,"IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987,Jul. 2002.
[49] Z. Sun, H. Zhang, T. Tan, and J. Wang, "Iris image classification basedon hierarchical visual codebook," IEEE Trans. Pattern Anal. Mach.Intell., vol. 36, no. 6, pp. 1120–1133, Jun. 2014.
[50] W. R. Schwartz, A. Rocha, and H. Pedrini, "Face spoofing detectionthrough partial least squares and low-level descriptors," in Proc. IEEEInt. Joint Conf. Biometrics (IJCB), Oct. 2011, pp. 1–8.
[51] D. Yi, Z. Lei, Z. Zhang, and S. Li, "Face anti-spoofing: Multi-spectralapproach," in Handbook of Biometric Anti-Spoofing (Advances in ComputerVision and Pattern Recognition), S. Marcel, M. S. Nixon, andS. Z. Li, Eds. London, U.K.: Springer-Verlag, 2014, pp. 83–102.
[52] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection fromsingle images using micro-texture analysis," in Proc. Int. Joint Conf.Biometrics (IJCB), Oct. 2011, pp. 1–7.
[53] A. Anjos and S. Marcel, "Counter-measures to photo attacks in facerecognition: A public database and a baseline," in Proc. Int. Joint Conf.Biometrics, Oct. 2011, pp. 1–7.
[54] T.-W. Lee, G.-H. Ju, H.-S. Liu, and Y.-S. Wu, "Liveness detection usingfrequency entropy of image sequences," in Proc. IEEE Int. Conf. Acoust.,Speech, Signal Process. (ICASSP), May 2013, pp. 2367–2370.
[55] A. da Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha, "Video-basedface spoofing detection through visual rhythm analysis," in Proc. 25thConf. Graph., Patterns Images (SIBGRAPI), 2012, pp. 221–228.
[56] N. Erdogmus and S. Marcel, "Spoofing 2D face recognition systemswith 3D masks," in Proc. Int. Conf. Biometrics Special InterestGroup (BIOSIG), Sep. 2013, pp. 1–8.
[57] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gaborbinary pattern histogram sequence (LGBPHS): A novel non-statisticalmodel for face representation and recognition," in Proc. 10th IEEE Int.Conf. Comput. Vis. (ICCV), vol. 1. Oct. 2005, pp. 786–791.
[58] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Facerecognition by elastic bunch graph matching," IEEE Trans. Pattern Anal.Mach. Intell., vol. 19, no. 7, pp. 775–779, Jul. 1997.
[59] M. Günther, D. Haufe, and R. P. Würtz, "Face recognition with disparitycorrected Gabor phase differences," in Proc. 22nd Int. Conf. Artif. NeuralNetw. Mach. Learn. (ICANN), 2012, pp. 411–418.
[60] N. Kose and J.-L. Dugelay, "On the vulnerability of face recognitionsystems to spoofing mask attacks," in Proc. IEEE Int. Conf. Acoust.,Speech, Signal Process. (ICASSP), May 2013, pp. 2357–2361.
[61] N. Kose and J.-L. Dugelay, "Countermeasure for the protection of facerecognition systems against mask attacks," in Proc. 10th IEEE Int. Conf.Workshops Autom. Face Gesture Recognit. (FG), Apr. 2013, pp. 1–6.
[62] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a singleimage with sparse low rank bilinear discriminative model," in Proc. 11thEur. Conf. Comput. Vis. (ECCV), 2010, pp. 504–517.
[63] N. Kose and J.-L. Dugelay, "Reflectance analysis based countermeasuretechnique to detect face mask attacks," in Proc. 18th Int. Conf. Digit.Signal Process. (DSP), 2013, pp. 1–6.
[64] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Canface anti-spoofing countermeasures work in a real world scenario?" inProc. IAPR Int. Conf. Biometrics (ICB), 2013, pp. 1–8.
[65] T. de Freitas Pereira et al., "Face liveness detection using dynamictexture," EURASIP J. Image Video Process., vol. 2014, p. 2, Jan. 2014.
[66] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia,"Fingerprint liveness detection based on quality measures," in Proc. Int.Conf. Biometrics, Identity, Secur. (BIdS), 2009, pp. 1–8.

[67] G. L. Marcialis et al., "First international fingerprint liveness detectioncompetition—LivDet 2009," in Proc. 15th Int. Conf. Image Anal.Process. (ICIAP) vol. 5716. 2009, pp. 12–23. [Online]. Available:http://prag.diee.unica.it/LivDet09

[68] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia,"A high performance fingerprint liveness detection method based onquality related features," Future Generat. Comput. Syst., vol. 28, no. 1,pp. 311–321, 2012.