

Deep Representations for Iris, Face, and Fingerprint Spoofing Detection

David Menotti, *Member, IEEE*, Giovani Chiachia, Allan Pinto, *Student Member, IEEE*,
William Robson Schwartz, *Member, IEEE*, Helio Pedrini, *Member, IEEE*,
Alexandre Xavier Falcão, *Member, IEEE*, and
Anderson Rocha, *Member, IEEE*

Abstract—Biometrics systems have significantly improved person identification and authentication, playing an important role in personal, national, and global security. However, these systems might be deceived (or spoofed) and, despite the recent advances in spoofing detection, current solutions often rely on domain knowledge, specific biometric reading systems, and attack types. We assume 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. This strategy not only provides better comprehension of how these approaches interplay, but also creates systems that exceed the best known results in eight out of the nine benchmarks. The results strongly indicate that spoofing detection systems based on convolutional networks can be robust to attacks already known and possibly adapted, with little effort, to image-based attacks that are yet to come.

Index Terms—Deep learning, convolutional networks, hyperparameter architecture optimization, filter weights learning, back-propagation, spoofing detection.

Manuscript received July 3, 2014; revised November 18, 2014 and January 19, 2015; accepted January 27, 2015. Date of publication February 2, 2015; date of current version March 20, 2015. This work was supported in part by the São Paulo Research Foundation under Grant 2010/05647-4, Grant 2011/22749-8, Grant 2013/04172-0, and Grant 2013/11359-0, in part by the Federal University of Ouro Preto through the Brazilian National Research Council under Grant 303673/2010-9, Grant 304352/2012-8, Grant 307113/2012-4, Grant 477662/2013-7, Grant 487529/2013-8, Grant 479070/2013-0, and Grant 477457/2013-4, in part by the Minas Gerais Research Foundation under Grant APQ-01806-13, and in part by the Coordination for the Improvement of Higher Education Personnel (CAPES) DeepEyes Project. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stan Z. Li. (David Menotti and Giovani Chiachia contributed equally to this work.)

D. Menotti was with the Institute of Computing, University of Campinas, Campinas 13083-852, Brazil. He is now with the Department of Computing, Federal University of Ouro Preto, Ouro Preto 35400-000, Brazil (e-mail: menottid@gmail.com).

G. Chiachia, A. Pinto, H. Pedrini, A. X. Falcão, and A. Rocha are with the Institute of Computing, University of Campinas, Campinas 13083-852, Brazil (e-mail: chiachia@ic.unicamp.br; allan.pinto@ic.unicamp.br; helio.pedrini@ic.unicamp.br; afalcao@ic.unicamp.br; anderson.rocha@ic.unicamp.br).

W. R. Schwartz is with the Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte 31270-010, Brazil (e-mail: william@dcc.ufmg.br).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIFS.2015.2398817

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.

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

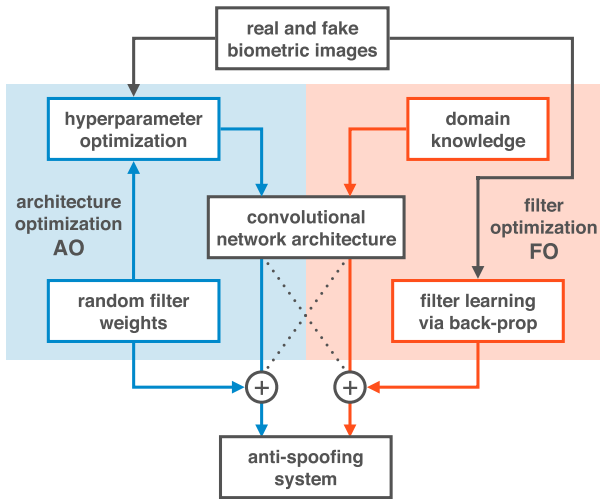


Fig. 1. Schematic diagram detailing how anti-spoofing systems are built from spoofing detection benchmarks. Architecture optimization (AO) is shown on the left and filter optimization (FO) on the right. In this work, we not only evaluate AO and FO in separate, but also in combination, as indicated by 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 hyperparameter 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 a 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 electroencephalogram (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 MobLive² [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 hard-coded 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

¹It also appears in a lecture of Daugman at IBC 2004 [29].

²MobLive 2014, Intl. Joint Conference on Biometrics (IJCB).

TABLE I
MAIN FEATURES OF THE BENCHMARKS CONSIDERED HEREIN

Modality	Benchmark/Dataset	Color	Dimension <i>cols</i> × <i>rows</i>	# 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	600	600	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			

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 problem by extracting meaningful features directly from the data regardless of the input type (image, video, or 3D masks).

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].

Graganiello 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 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

1) *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.

2) *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.

3) *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

1) *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: (1) *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; (2) *mobile attack*, which presents to the acquisition sensor photos and videos taken with an iPhone using the iPhone screen; and (3) *high-definition attack*, in which high resolution photos and videos taken with an iPad are presented to the acquisition sensor using the iPad screen.

2) *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. The 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

1) *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

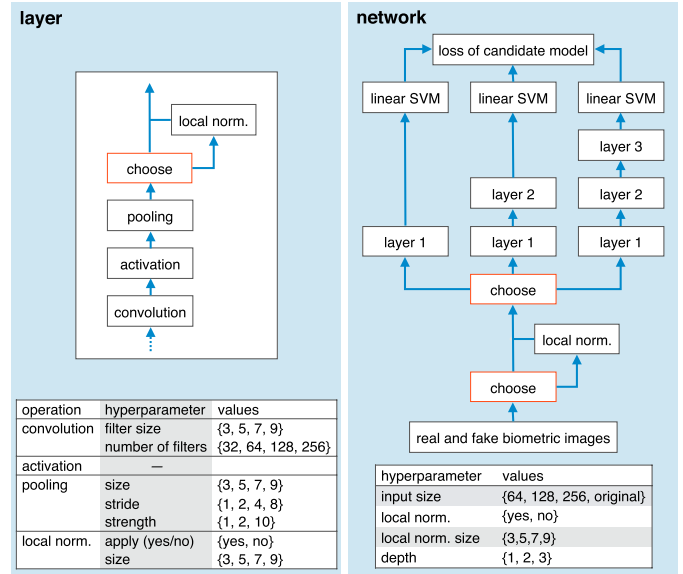


Fig. 2. Schematic diagram for architecture optimization (AO) illustrating how operations are stacked in a layer (left) and how the network is instantiated and evaluated according to possible hyperparameter values (right). Note that a three-layered convolutional network of this type has a total of 25 hyperparameters governing both its architecture and its overall behaviour through a particular instance of stacked operations.

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.

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 hyperparameter 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” hyperparameter. In fact, there are other six hyperparameters, 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 hyperparameter, 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 hyperparameters, such as the size of the input image, that expand possibilities to a myriad potential architectures.

The overall set of possible hyperparameter values is called *search space*, which in this case is discrete and contains variables that are only meaningful in combination with others. For example, hyperparameters 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 an 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 hyperparameters, 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:

- 1) Randomly — and uniformly, in our case — sample values from the hyperparameter *search space*;
- 2) Extract features from real and fake training images with the candidate architecture;
- 3) Evaluate the architecture according to an *optimization objective* based on linear SVM scores;
- 4) Repeat steps 1–3 until a *termination criterion* is met;
- 5) 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 ($C=10^5$) 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.

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].

³<http://www.cs.toronto.edu/~kriz/cifar.html>.

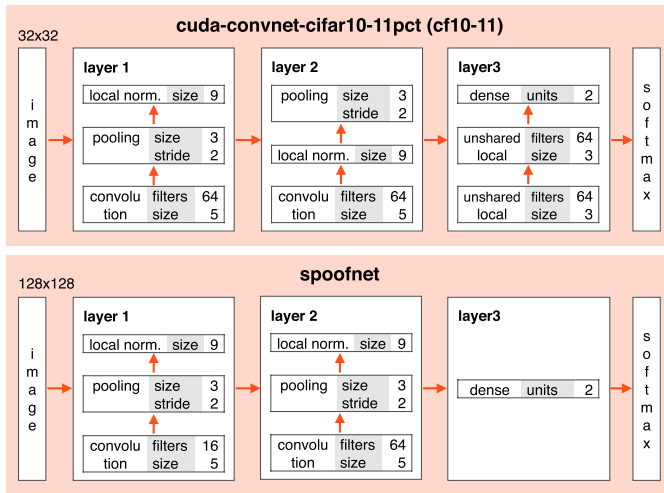


Fig. 3. Architecture of convolutional network found in the Cuda-convnet library [80] and here used as reference for filter optimization (*cf10-11*, top). Proposed network architecture extending upon *cf10-11* to better suit spoofing detection problems (*spooonet*, bottom). Both architectures are typical 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:

- 1) 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;
- 2) For another 40 epochs, resume training now considering all four batches for training;
- 3) Reduce the learning rate by a factor of 10, and train the network for another 10 epochs;
- 4) 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 architecture for spoofing detection that we call *spooonet*. 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 *spooonet*, and we believe that it might enable the network to be more sensitive to subtle local patterns in the images.

⁴<https://code.google.com/p/cuda-convnet/wiki/Methodology>.

TABLE II
INPUT IMAGE DIMENSIONALITY AFTER BASIC PREPROCESSING
ON FACE AND FINGERPRINT IMAGES (HIGHLIGHTED).
SEE SECTION IV-C FOR DETAILS

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

In order to train *spooonet*, 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 overfitting, 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 *spooonet*, 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.

1) *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.

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

- (a) *Biometrika*: We cropped the central region of size in columns and rows corresponding to 70% of the original image dimensions.
- (b) *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.
- (c) *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:

- 1) Retrieving prediction scores from the testing samples;
- 2) Calculating a threshold τ above which samples are predicted as attacks;
- 3) 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 hyperparameters 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.

V. EXPERIMENTS AND RESULTS

In this section, we evaluate the effectiveness of the proposed methods for spoofing detection. We show experiments for the architecture optimization and filter learning approaches along with their combination for detecting iris, face, and fingerprint spoofing on the nine benchmarks described in Section III. We also present results for the *spoofnet*, which incorporates some domain-knowledge on the problem. We compare all of the results with the state-of-the-art counterparts. Finally, we discuss the pros and cons of using such approaches and their combination along with efforts to understand the type of features learned and some efficiency questions when testing the proposed methods.

A. Architecture Optimization (AO)

Table III presents AO results in detail as well as previous state-of-the-art (SOTA) performance for the considered benchmarks. With this approach, we can outperform four SOTA methods in all three biometric modalities. Given that AO assumes little knowledge about the problem domain, this is remarkable. Moreover, performance is on par in other four benchmarks, with the only exception of Swipe. Still in Table III, we can see information about the best architecture such as time taken to evaluate it (feature extraction + 10-fold validation), input size, depth, and dimensionality of the output representation in terms of $columns \times rows \times feature\ maps$.

Regarding the number of layers in the best architectures, we can observe that six out of nine networks use two layers, and three use three layers. We speculate that the number of layers obtained is a function of the problem complexity. In fact, even though there are many other hyperparameters involved, the number of layers play an important role in this issue, since it directly influences the level of non-linearity and abstraction of the output with respect to the input.

⁵<http://scikit-learn.org>.

TABLE III

OVERALL RESULTS CONSIDERING RELEVANT INFORMATION OF THE BEST FOUND ARCHITECTURES, DETECTION ACCURACY (ACC) AND HTER VALUES ACCORDING TO THE EVALUATION PROTOCOL, AND STATE-OF-THE-ART (SOTA) PERFORMANCE

modality	benchmark	architecture optimization (AO)					our results		SOTA results		
		time (secs.)	size (pixels)	layers (#)	features (#)	objective (%)	ACC (%)	HTER (%)	ACC (%)	HTER (%)	Ref.
iris	Warsaw	52+35	640	2	$10 \times 15 \times 64$ (9600)	98.21	99.84	0.16	97.50	—	[26]
	Biosec	80+34	640	3	$2 \times 5 \times 256$ (2560)	97.56	98.93	1.17	100.00	—	[38]
	MobBIOfake	18+37	250	2	$5 \times 7 \times 256$ (8960)	98.94	98.63	1.38	99.75	—	[6]
face	Replay-Attack	69+15	256	2	$3 \times 3 \times 256$ (2304)	94.65	98.75	0.75	—	5.11	[88]
	3DMAD	55+15	128	2	$5 \times 5 \times 64$ (1600)	98.68	100.00	0.00	—	0.95	[56]
fingerprint	Biometrika	66+25	256	2	$2 \times 2 \times 256$ (1024)	90.11	96.50	3.50	98.30	—	[11]
	Crossmatch	112+12	675	3	$2 \times 3 \times 256$ (1536)	91.70	92.09	8.44	68.80	—	[11]
	Italdata	46+27	432	3	$16 \times 13 \times 128$ (26624)	86.89	97.45	2.55	99.40	—	[11]
	Swipe	97+51	962	2	$53 \times 3 \times 32$ (5088)	90.32	88.94	11.47	96.47	—	[11]

With respect to the input size, we can see in comparison with Table II, that the best performing architectures often use the original image size. This was the case for all iris benchmarks and for three (out of four) fingerprint benchmarks. For face benchmarks, a larger input was preferred for Replay-Attack, while a smaller input was preferred for 3DMAD. We hypothesize that this is also related to the problem difficulty, given that Replay-Attack seems to be more difficult, and that larger inputs tend to lead to larger networks.

We still notice that the dimensionality of the obtained representations are, in general, smaller than 10K features, except for Italdata. Moreover, for the face and iris benchmarks, it is possible to roughly observe a relationship between the optimization objective calculated in the training set and the detection accuracy measure on the testing set (Section IV-D), which indicates the appropriateness of the objective for these tasks. However, for the fingerprint benchmarks, this relationship does not exist, and we accredit this to either a deficiency of the optimization objective in modelling these problems or to the existence of artifacts in the training set misguiding the optimization.

B. Filter Optimization (FO)

Table IV shows the results for FO, where we repeat architecture optimization (AO) results (with random filters) in the first column to facilitate comparisons. Overall, we can see that both networks, *cf10-11* and *spoofnnet* have similar behavior across the biometric modalities.

Surprisingly, *cf10-11* obtains excellent performance in all four fingerprint benchmarks as well as in the MobBIOFake, exceeding SOTA in three cases, in spite of the fact that it was used without any modification. However, in both face problems and in two iris problems, *cf10-11* performed poorly. Such difference in performance was not possible to anticipate by observing training errors, which steadily decreased in all cases until training was stopped. Therefore, we believe that in these cases FO was misguided by the lack of training data or structure in the training samples irrelevant to the problem.

To reinforce this claim, we performed experiments with filter optimization (FO) in *spoofnnet* by varying the training set size with 20%, 40%, and 50% of fingerprint benchmarks. As expected, in all cases, the less training examples, the worse

TABLE IV

RESULTS FOR FILTER OPTIMIZATION (FO) IN *cf10-11* AND *spoofnnet* (FIG. 3). EVEN THOUGH BOTH NETWORKS PRESENT SIMILAR BEHAVIOR, *spoofnnet* IS ABLE TO PUSH PERFORMANCE EVEN FURTHER IN PROBLEMS WHICH *cf10-11* WAS ALREADY GOOD FOR. ARCHITECTURE OPTIMIZATION (AO) RESULTS (WITH RANDOM FILTERS) ARE SHOWN IN THE FIRST COLUMN TO FACILITATE COMPARISONS

modality (metric)	benchmark	filter			SOTA
		random AO	<i>cf10-11</i>	<i>spoofnnet</i>	
iris (ACC)	Warsaw	99.84	67.20	66.42	97.50
	Biosec	98.93	59.08	47.67	100.00
	MobBIOfake	98.63	99.13	100.00	99.75
face (HTER)	Replay-Attack	0.75	55.13	55.38	5.11
	3DMAD	0.00	40.00	24.00	0.95
fingerprint (ACC)	Biometrika	96.50	98.50	99.85	98.30
	Crossmatch	92.09	97.33	98.23	68.80
	Italdata	97.45	97.35	99.95	99.40
	Swipe	88.94	98.70	99.08	96.47

is the generalization of the *spoofnnet* (lower classification accuracies). Considering the training phase, for instance, when using 50% of training set or less, the accuracy achieved by the learned representation is far worse than the one achieved when using 100% of training data. This fact reinforces the conclusion presented herein regarding the small sample size problem. Maybe a fine-tuning of some parameters, such as the number of training epochs and the learning rates, can diminish the impact of the small sample size problem stated here, however, this is an open research topic by itself.

For *spoofnnet*, the outcome is similar. As we expected, the proposed architecture was able to push performance even further in problems which *cf10-11* was already good for, outperforming SOTA in five out of nine benchmarks. This is possibly because we made the *spoofnnet* architecture simpler, with less parameters, and taking input images with a size better suited to the problem.

As compared to the results in AO, we can observe a good balance between the approaches. In AO, the resulting convolutional networks are remarkable in the face benchmarks. In FO, networks are remarkable in fingerprint problems. While in AO all optimized architectures have good performance

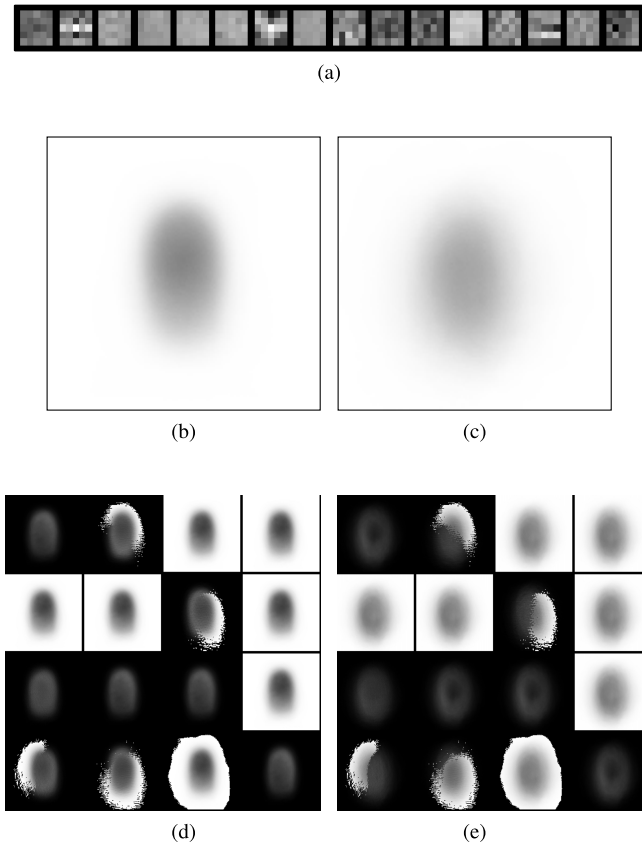


Fig. 4. Activation maps of the filters that compose the first convolutional layer when forwarding real and fake images through the network. (a) Filter weights of the first convolutional layer that were learned using the backpropagation algorithm. (b) Mean for real images (test set). (c) Mean for fake images (test set). (d) Activation maps for real images. (e) Activation maps for fake images.

in iris problems, FO excelled in one of these problems, MobBIOFake, with a classification accuracy of 100%. In general, AO seems to result in convolutional networks that are more stable across the benchmarks, while FO shines in problems in which learning effectively occurs. Considering both AO and FO, we can see in Table IV that we outperformed SOTA methods in eight out of nine benchmarks. The only benchmark where SOTA performance was not achieved is Biosec, but even in this case the result obtained with AO is competitive.

Understanding how a set of deep learned features capture properties and nuances of a problem is still an open question in the vision community. However, in an attempt to understand the behavior of the operations applied onto images after they are forwarded through the first convolutional layer, we generate Fig. 4a that illustrates the filters learned via backpropagation algorithm and Figs. 4b and 4c showing the mean of real and fake images that compose the test set, respectively. To obtain output values from the first convolutional layer and get a sense of them, we also instrumented the *spoofnet* convolutional network to forward the real and fake images from the test set through network. Figs 4d and 4e show such images for the real and fake classes, respectively.

We can see in Fig. 4a that the filters learned patterns resemble textural patterns instead of edge patterns as usually

occurs in several computer vision problems [13], [15]. This is particularly interesting and in line with several anti-spoofing methods in the literature which also report good results when exploring texture information [11], [52].

In addition, Fig. 4b and 4c show there are differences between real and fake images from test, although apparently small in such a way that a direct analysis of the images would not be enough for decision making. However, when we analyze the mean activation maps for each class, we can see more interesting patterns. In Figs. 4d and 4e, we have sixteen pictures with 128×128 pixel resolution. These images correspond to the sixteen filters that composing the first layer of the *spoofnet*. Each position (x, y) in these 128×128 images corresponds to a 5×5 area (receptive field units) in the input images. Null values in a given unit means that the receptive field of the unit was not able to respond to the input stimuli. In contrast, non-null values mean that the receptive field of the unit had a responsiveness to the input stimuli.

We can see that six filters have a high responsiveness to the background information of the input images (filters predominantly white) whilst ten filters did not respond to background information (filters predominantly black). From left to right, top to bottom, we can see also that the images corresponding to the filters 2, 7, 13, 14 and 15 have high responsiveness to information surrounding the central region of the sensor (usually where fingerprints are present) and rich in texture details. Although these regions of high and low responsiveness are similar for both classes we can notice some differences. A significant difference in this first convolutional layer to images for the different classes is that the response of the filters regarding to fake images (Fig 4e) generates a blurring pattern, unlike the responses of the filters regarding to real images (Fig 4d) which generate a sharper pattern. We believe that the same way as the first layer of a convolutional network has the ability to respond to simple and relevant patterns (edge information) to a problem of recognition objects in general, in computer vision, the first layer in the *spoofnet* also was able to react to a simple pattern recurrent in spoof problems, the blurring effect, an artifact previously explored in the literature [76]. Finally, we are exploring visualisation only of the first layer; subsequent layers of the network can find new patterns in these regions activated by the first layer further emphasizing class differences.

C. Interplay Between AO and FO

In the previous experiments, architecture optimization (AO) was evaluated using random filters and filter optimization (FO) was carried out in the predefined architectures *cf10-11* and *spoofnet*. A natural question that emerges in this context is how these methods would perform if we (i) combine AO and FO and if we (ii) consider random filters in *cf10-11* and *spoofnet*.

Results from these combinations are available in Table V and show a clear pattern. When combined with AO, FO again exceeds previous SOTA in all fingerprint benchmarks and performs remarkably good in MobBIOFake. However, the same difficulty found by FO in previous experiments for both face and two iris benchmarks is also observed here.

TABLE V

RESULTS FOR ARCHITECTURE AND FILTER OPTIMIZATION (AO+FO) ALONG WITH *cf10-11* AND *spoofnet* NETWORKS CONSIDERING RANDOM WEIGHTS. AO+FO SHOW COMPELLING RESULTS FOR FINGERPRINTS AND ONE IRIS BENCHMARK (MOBBIOFAKE). WE CAN ALSO SEE THAT *spoofnet* CAN BENEFIT FROM RANDOM FILTERS IN SITUATIONS IT WAS NOT GOOD FOR WHEN USING FILTER LEARNING (e.g., REPLAY-ATTACK)

modality (metric)	benchmark	filter			
		optimized	random		SOTA
		AO	<i>cf10-11</i>	<i>spoofnet</i>	
(iris ACC)	Warsaw	59.55	87.06	96.44	97.50
	Biosec	57.50	97.33	97.42	100.00
	MobBIOfake	99.38	77.00	72.00	99.75
(face HTER)	Replay-Attack	55.88	5.62	3.50	5.11
	3DMAD	40.00	8.00	4.00	0.95
(fingerprint ACC)	Biometrika	99.30	77.45	94.70	98.30
	Crossmatch	98.04	83.11	87.82	68.80
	Italdata	99.45	76.45	91.05	99.40
	Swipe	99.08	87.60	96.75	96.47

Even though *spoofnet* performs slightly better than AO in the cases where SOTA is exceeded (Table IV), it is important to remark that our AO approach may result in architectures with a much larger number of filter weights to be optimized, and this may have benefited *spoofnet*.

It is also interesting to observe in Table V the results obtained with the use of random filters in *cf10-11* and *spoofnet*. The overall balance in performance of both networks across the benchmarks is improved, similar to what we have observed with the use of random filters in Table III. An striking observation is that *spoofnet* with random filters exceed previous SOTA in Replay-Attack, and this supports the idea that the poor performance of *spoofnet* in Replay-Attack observed in the FO experiments (Table IV) was not a matter of architecture.

D. Runtime

We estimate time requirements for anti-spoofing systems built with convolutional networks based on measurements obtained in architecture optimization (AO). We can see in Table III that the most computationally intensive deep representation is the one found for the Swipe benchmark, and demands 148 (97+51) seconds to process 2,200 images. Such a running time is only possible due to the GPU+CPU implementation used (Section IV-E), which is critical for this type of learning task. In a hypothetical operational scenario, we could ignore the time required for classifier training (51 seconds, in this case). Therefore, we can estimate that, on average, a single image captured by a Swipe sensor would require approximately 45 milliseconds — plus a little overhead — to be fully processed in this hypothetical system. Moreover, the existence of much larger convolutional networks running in realtime in budgeted mobile devices [89] also supports the idea that the approach is readily applicable in a number of possible scenarios.

E. Visual Assessment

In Fig. 5, we show examples of hit and missed testing samples lying closest to the real-fake decision boundary of the best performing system in each benchmark. A magnified visual inspection on these images may give us some hint about properties of the problem to which the learned representations are sensitive.

While it is difficult to infer anything concrete, it is interesting to see that the real missed sample in Biosec is quite bright, and that skin texture is almost absent in this case. Still, we may argue that a noticeable difference exists in Warsaw between the resolution used to print the images that led to the fake hit and the fake miss.

Regarding the face benchmarks, the only noticeable observation from Replay-Attack is that the same person is missed both when providing to the system a real and a fake biometric reading. This may indicate that some individuals are more likely to successfully attack a face recognition systems than others. In 3DMAD, it is easy to see the difference between the real and fake hits. Notice that there was no misses in this benchmark.

A similar visual inspection is much harder in the fingerprint benchmarks, even though the learned deep representations could effectively characterize these problems. The only observation possible to be made here is related to the fake hit on CrossMatch, which is clearly abnormal. The images captured with the Swipe sensor are naturally narrow and distorted due to the process of acquisition, and this distortion prevents any such observation.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we investigated two deep representation research approaches for detecting spoofing in different biometric modalities. On one hand, we 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, we sought to learn 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. In addition, we incorporated our experience with architecture optimization as well as with training filter weight for a given architecture into a more interesting and adapted network, *spoofnet*.

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. However, in this case, it is possible to achieve a 98.93% against 100.0% accuracy of the literature. 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 is a promising venue. Another indication of this comes from the initial study we did for understanding the type of filters generated by the learning process. Considering the fingerprint

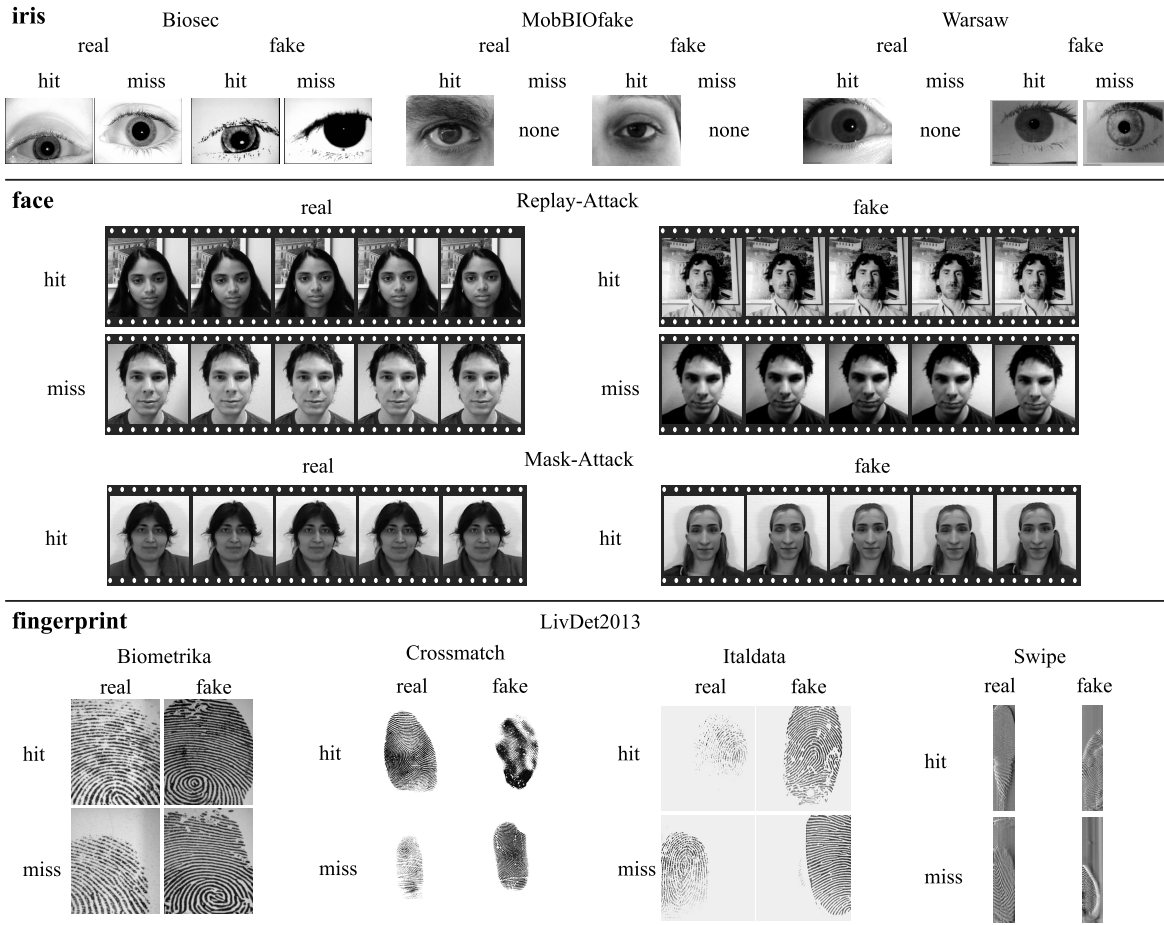


Fig. 5. Examples of hit and missed testing samples lying closest to the real-fake decision boundary of each benchmark. A magnified visual inspection on these images may suggest some properties of the problem to which the learned representations are sensitive.

case, learning directly from data, it was possible to come up with discriminative filters that explore the blurring artifacts due to recapture. This is particularly interesting as it is in line with previous studies using custom-tailored solutions [76].

It is important to emphasise the interplay between the architecture and filter optimization approaches for the spoofing problem. It is well-known in the deep learning literature that when thousands of samples are available for learning, the filter learning approach is a promising path. Indeed, we could corroborate this through fingerprint benchmarks that considers a few thousand samples for training. However, it was not the case for faces and two iris benchmarks which suffer from the small sample size problem (SSS) and subject variability hindering the filter learning process. In these cases, the architecture optimization approach was able to learn representative and discriminative features providing comparable spoofing effectiveness to the SOTA results in almost all benchmarks, and specially outperforming them in three out of four SOTA results when the filter learning approach failed. It is worth mentioning that sometimes it is still possible to learn meaningful features from the data even with a small sample size for training. We believe this happens in more well-posed datasets with less variability between training/testing data as it is the case of MobioBIOfake benchmark in which the AO approach achieved 99.38% just 0.37% behind the SOTA result.

As the data tell it all, the decision to which path to follow can also come from the data. Using the evaluation/validation set during training, the researcher/developer can opt for optimizing architectures, learn filters or both. If training time is an issue and a solution must be presented overnight, it might be interesting to consider an already learned network that incorporates some additional knowledge in its design. In this sense, *spoofnet* could be a good choice. In all cases, if the developer can incorporate more training examples, the approaches might benefit from such augmented training data.

The proposed approaches can also be adapted to other biometric modalities not directly dealt with herein. The most important difference would be in the input type of data since all discussed solutions directly learn their representations from the data.

For the case of iris spoofing detection, here we dealt only with iris spoofing printed attacks and some experimental datasets using cosmetic contact lenses have recently become available allowing researchers to study this specific type of spoofing [7], [8]. For future work, we intend to evaluate such datasets using the proposed approaches here and also consider other biometric modalities such as palm, vein, and gait.

Finally, it is important to take all the results discussed herein with a grain of salt. We are not presenting the final word in spoofing detection. In fact, there are important

additional research that could finally take this research another step forward. We envision the application of deep learning representations on top of pre-processed image feature maps (e.g., LBP-like feature maps, acquisition-based maps exploring noise signatures, visual rhythm representations, etc.). With an n -layer feature representation, we might be able to explore features otherwise not possible using the raw data. In addition, exploring temporal coherence and fusion would be also important for video-based attacks.

APPENDIX CONVOLUTIONAL NETWORK OPERATIONS

Our networks use classic convolutional operations that can be viewed as linear and non-linear image processing operations. When stacked, these operations essentially extract higher level representations, named *multiband images*, whose pixel attributes are concatenated into high-dimensional feature vectors for later pattern recognition.⁶

Assuming $\hat{I} = (D_I, \vec{I})$ as a multiband image, where $D_I \subset Z^2$ is the image domain and $\vec{I}(p) = \{I_1(p), I_2(p), \dots, I_m(p)\}$ is the attribute vector of a m -band pixel $p = (x_p, y_p) \in D_I$, the aforementioned operations can be described as follows.

1) *Filter Bank Convolution*: Let $\mathcal{A}(p)$ be a squared region centered at p of size $L_A \times L_A$, such that $\mathcal{A} \subset D_I$ and $q \in \mathcal{A}(p)$ iff $\max(|x_q - x_p|, |y_q - y_p|) \leq (L_A - 1)/2$. Additionally, let $\Phi = (\mathcal{A}, W)$ be a filter with weights $W(q)$ associated with pixels $q \in \mathcal{A}(p)$. In the case of multiband filters, filter weights can be represented as vectors $\vec{W}_i(q) = \{w_{i,1}(q), w_{i,2}(q), \dots, w_{i,m}(q)\}$ for each filter i of the bank, and a multiband filter bank $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_n\}$ is a set of filters $\Phi_i = (\mathcal{A}, \vec{W}_i)$, $i = \{1, 2, \dots, n\}$.

The convolution between an input image \hat{I} and a filter Φ_i produces a band i of the filtered image $\hat{J} = (D_J, \vec{J})$, where $D_J \subset D_I$ and $\vec{J} = (J_1, J_2, \dots, J_n)$, such that for each $p \in D_J$,

$$J_i(p) = \sum_{\forall q \in \mathcal{A}(p)} \vec{I}(q) \cdot \vec{W}_i(q). \quad (1)$$

2) *Rectified Linear Activation*: Filter activation in this work is performed by rectified linear units (RELU) of the type present in many state-of-the-art convolutional architectures [13], [21] and is defined as

$$J_i(p) = \max(J_i(p), 0). \quad (2)$$

3) *Spatial Pooling*: Spatial pooling is an operation of paramount importance in the literature of convolutional networks [19] that aims at bringing translational invariance to the features by aggregating activations from the same filter in a given region.

Let $\mathcal{B}(p)$ be a pooling region of size $L_B \times L_B$ centered at pixel p and $D_K = D_J/s$ be a regular subsampling of every s pixels $p \in D_J$. We call s the *stride* of the pooling operation. Given that $D_J \subset Z^2$, if $s = 2$, $|D_K| = |D_J|/4$, for example.

⁶This appendix describes convolutional networks from an image processing perspective, therefore the use of terms like image *domain*, image *band*, etc.

The pooling operation resulting in the image $\hat{K} = (D_K, \vec{K})$ is defined as

$$K_i(p) = \sqrt[\alpha]{\sum_{\forall q \in \mathcal{B}(p)} J_i(q)^\alpha}, \quad (3)$$

where $p \in D_K$ are pixels in the new image, $i = \{1, 2, \dots, n\}$ are the image bands, and α is a hyperparameter that controls the sensitivity of the operation. In other words, our pooling operation is the L_α -norm of values in $\mathcal{B}(p)$. The stride s and the size of the pooling neighborhood defined by L_B are other hyperparameters of the operation.

4) *Divisive Normalization*: The last operation considered in this work is divisive normalization, a mechanism widely used in top-performing convolutional networks [13], [21] that is based on gain control mechanisms found in cortical neurons [90].

This operation is also defined within a squared region $\mathcal{C}(p)$ of size $L_C \times L_C$ centered at pixel p such that

$$O_i(p) = \frac{K_i(p)}{\sqrt{\sum_{j=1}^n \sum_{\forall q \in \mathcal{C}(p)} K_j(q)^2}} \quad (4)$$

for each pixel $p \in D_O \subset D_K$ of the resulting image $\hat{O} = (D_O, \vec{O})$. Divisive normalization promotes competition among pooled filter bands such that high responses will prevail even more over low ones, further strengthening the robustness of the output representation \vec{O} .

ACKNOWLEDGMENT

D. Menotti thanks FAPESP for a grant to acquiring two NVIDIA GeForce GTX Titan Black with 6GB each. The authors also thank NVIDIA for donating five GPUs used in the experiments, a Tesla K40 with 12GB to A. X. Falcão, two GeForce GTX 680 with 2GB each to G. Chiachia, and two GeForce GTX Titan Black with 6GB each to D. Menotti.

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 hill-climbing 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. [Online]. Available: http://atvs.ii.uam.es/files/2007_SWB_VulnerabilitiesRecentAdvances_Galbally.pdf
- [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.inesporto.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. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran & Associates Inc., 2012.
- [14] D. C. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep 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: Closing the gap to human-level performance in face verification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.
- [17] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A high-throughput screening approach to discovering good forms of biologically inspired visual 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-based learning 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. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1–8.
- [21] D. D. Cox and N. Pinto, "Beyond simple features: A large-scale feature search 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, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing 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 of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 115–123.
- [24] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks 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 images in iris verification," in *Proc. 1st Eur. Workshop Biometrics Identity Manage. (BioID)*, vol. 5372. 2008, pp. 181–190.
- [26] A. Czajka, "Database of iris printouts and its application: Development of 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," in *Proc. 7th Int. Biometrics Conf.*, 2004.
- [30] E. C. Lee, K. R. Park, and J. Kim, "Fake iris detection by using purkinje image," 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," in *Proc. 40th Annu. IEEE Int. Carnahan Conf. Secur. Technol.*, 2006, pp. 122–129.
- [32] M. Kanematsu, H. Takano, and K. Nakamura, "Highly reliable liveness detection 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 based on 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 recognition with 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 accuracy of 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 experimental study of pupil constriction for liveness detection," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 252–258.
- [37] T. Kathikeyan and B. Sabarigiri, "Countermeasures against IRIS spoofing and 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 liveness detection based on quality related features," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, 2012, pp. 271–276.
- [39] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.
- [40] J. Fierrez-Aguilar, J. Ortega-Garcia, D. Torre-Toledano, and J. Gonzalez-Rodriguez, "Biosec baseline corpus: A multimodal biometric database," *Pattern Recognit.*, vol. 40, no. 4, pp. 1389–1392, 2007.
- [41] A. F. Sequeira, J. Murari, and J. S. Cardoso, "Iris liveness detection methods in mobile applications," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (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 detection methods 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, "Robust iris localisation in challenging scenarios," in *Computer Vision, Imaging and Computer Graphics: Theory and Applications (Communications in Computer and Information Science)*. Berlin, Germany: Springer-Verlag, 2004.
- [45] J. Doyle and K. W. Bowyer. (2014). *Notre Dame Image Database for Contact 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 classification using local phase quantization," in *Proc. 3rd Int. Conf. Image Signal Process. (ICISP)*, 2008, pp. 236–243.
- [47] Z. Zhang, Z. Zhou, and H. Li, "Binary Gabor pattern: An efficient and robust 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. Maenpää, "Multiresolution gray-scale and 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 based on 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 detection through partial least squares and low-level descriptors," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2011, pp. 1–8.
- [51] D. Yi, Z. Lei, Z. Zhang, and S. Li, "Face anti-spoofing: Multi-spectral approach," in *Handbook of Biometric Anti-Spoofing (Advances in Computer Vision and Pattern Recognition)*, S. Marcel, M. S. Nixon, and S. Z. Li, Eds. London, U.K.: Springer-Verlag, 2014, pp. 83–102.
- [52] J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single 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 face recognition: 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 using frequency 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-based face spoofing detection through visual rhythm analysis," in *Proc. 25th Conf. Graph., Patterns Images (SIBGRAPI)*, 2012, pp. 221–228.

- [56] N. Erdogmus and S. Marcel, "Spoofing 2D face recognition systems with 3D masks," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2013, pp. 1–8.
- [57] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model 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, "Face recognition 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 disparity corrected Gabor phase differences," in *Proc. 22nd Int. Conf. Artif. Neural Netw. Mach. Learn. (ICANN)*, 2012, pp. 411–418.
- [60] N. Kose and J.-L. Dugelay, "On the vulnerability of face recognition systems 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 face recognition 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 single image with sparse low rank bilinear discriminative model," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 504–517.
- [63] N. Kose and J.-L. Dugelay, "Reflectance analysis based countermeasure technique 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, "Can face anti-spoofing countermeasures work in a real world scenario?" in *Proc. IAPR Int. Conf. Biometrics (ICB)*, 2013, pp. 1–8.
- [65] T. de Freitas Pereira *et al.*, "Face liveness detection using dynamic texture," *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 detection competition—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 on quality related features," *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 311–321, 2012.
- [69] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantization," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 537–540.
- [70] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, "LivDet 2011—Fingerprint liveness detection competition 2011," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, 2012, pp. 208–215. [Online]. Available: <http://people.clarkson.edu/projects/biosal/fingerprint/index.phps>
- [71] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Fingerprint liveness detection based on webner local image descriptor," in *Proc. IEEE Workshop Biometric Meas. Syst. Secur. Med. Appl.*, Sep. 2013, pp. 46–50.
- [72] X. Jia *et al.*, "Multi-scale block local ternary patterns for fingerprints vitality detection," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, 2013, pp. 1–6.
- [73] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection using binarized statistical image features," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl., Syst. (BTAS)*, Sep./Oct. 2013, pp. 1–6.
- [74] J. Kannala and E. Rahtu, "BSIF: Binarized statistical image features," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 1363–1366.
- [75] A. Hyvärinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, 1st ed. New York, NY, USA: Springer-Verlag, 2009.
- [76] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 710–724, Feb. 2014.
- [77] M. Chakka *et al.*, "Competition on counter measures to 2-D facial spoofing attacks," in *Proc. Int. Joint Conf. Biometrics (IJCB)*, Oct. 2011, pp. 1–6.
- [78] I. Chingovska *et al.*, "The 2nd competition on counter measures to 2D face spoofing attacks," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, Oct. 2013, pp. 1–6.
- [79] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2146–2153.
- [80] A. Krizhevsky. (2012). *CUDA-Cnnnet: High-Performance C++/CUDA Implementation of Convolutional Neural Networks*. [Online]. Available: <https://code.google.com/p/cuda-cnnnet/>
- [81] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 818–833.
- [82] P. Viola and M. J. Jones, "Robust real-time object detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2001.
- [83] A. A. Ross, K. Nandakumar, and A. K. Jain, "Score level fusion," in *Handbook of Multibiometrics* (International Series on Biometrics), vol. 6. New York, NY, USA: Springer-Verlag, 2006, pp. 91–142.
- [84] J. Bergstra, D. Tamins, and N. Pinto. (2013). *Hyperparameter Optimization for Convolutional Vision Architecture*. [Online]. Available: <https://github.com/hyperopt/hyperopt-cnn>
- [85] J. Bergstra *et al.*, "Theano: A CPU and GPU math compiler in python," in *Proc. 9th Python Sci. Comput. Conf. (SciPy)*, 2010, pp. 1–7.
- [86] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. ID 27.
- [87] G. Chiachia. (2014). *Hyperparameter Optimization Made Simple*. [Online]. Available: <https://github.com/giovanichiachia/simple-hp>
- [88] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel, "Complementary countermeasures for detecting scenic face spoofing attacks," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–7.
- [89] P. Wardern. (2014). *The SDK for Jetpac's iOS Deep Belief Image Recognition Framework*. [Online]. Available: <https://github.com/jetpacapp/DeepBeliefSDK>
- [90] W. S. Geisler and D. G. Albrecht, "Cortical neurons: Isolation of contrast gain control," *Vis. Res.*, vol. 32, no. 8, pp. 1409–1410, 1992.



David Menotti received the master's degrees in computer engineering and applied informatics from the Pontificia Universidade Católica do Paraná, Curitiba, Brazil, in 2001 and 2003, respectively, and the Ph.D. degree in computer science under a co-tutelage agreement from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, and the Université Paris-Est/Groupe ESIEE, Paris, France, in 2008. He has been an Associate Professor with the Department of Computing, Federal University of Ouro Preto (UFOP), Ouro Preto, Brazil, since 2008. From 2013 to 2014, he was on his sabbatical year with the Institute of Computing, University of Campinas, Campinas, Brazil, as a Post-Doctoral Researcher/Collaborator Researcher supported by FAPESP (grant 2013/4172-0). He is currently a Permanent and Collaborator Professor with the Post-Graduate Program in Computer Science, Department of Computing, UFOP, and the Department of Computer Science, UFMG, respectively.

His research interests include image processing, pattern recognition, computer vision, and information retrieval.



Giovanni Chiachia is currently a Post-Doctoral Researcher with the University of Campinas, Campinas, Brazil. His main research interest is to approach computer vision and other artificial intelligence tasks with brain-inspired machine learning techniques. He received the B.Sc. and M.S. degrees from São Paulo State University, São Paulo, Brazil, in 2005 and 2009, respectively, and the Ph.D. degree in computer science from the University of Campinas, in 2013. He held research scholar positions with the Fraunhofer Institute for Integrated Circuits, Erlangen, Germany, and Harvard University, Cambridge, MA, USA, as part of his graduate work, and was awarded first place for the performance of his face recognition system in the IEEE International Conference on Biometrics (2013). Beyond his research experience, he has a large experience in the IT industry and managing and being part of teams working on projects from a wide range of complexities and scales.



Allan Pinto received the B.Sc. degree from the University of São Paulo, São Paulo, Brazil, in 2011, and the M.Sc. degree from the University of Campinas, Campinas, Brazil, in 2013, where he is currently pursuing the Ph.D. degree with the Institute of Computing, all in computer science.

His research focuses on the areas of image and video analysis, computer forensics, pattern recognition, and computer vision in general, with a particular interest in spoofing detection in biometric systems.



William Robson Schwartz received the Ph.D. degree from the University of Maryland, College Park, MD, USA, and the B.Sc. and M.Sc. degrees from the Federal University of Parana, Curitiba, Brazil, all in computer science. He is currently a Professor with the Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil.

His research interests include computer vision, computer forensics, biometrics, and image processing.



Helio Pedrini received the Ph.D. degree in electrical and computer engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, and the M.Sc. degree in electrical engineering and the B.Sc. degree in computer science from the University of Campinas (Unicamp), Campinas, Brazil. He is currently a Professor with the Institute of Computing, Unicamp.

His research interests include image processing, computer vision, pattern recognition, machine learning, computer graphics, and computational geometry.



Alexandre Xavier Falcão is currently a Full Professor with the Institute of Computing, University of Campinas (Unicamp), Campinas, Brazil. He received the B.Sc. degree in electrical engineering from the Federal University of Pernambuco, Recife, Brazil, in 1988. He has been involved in biomedical image processing, visualization, and analysis, since 1991. He received the M.Sc. degree in electrical engineering from Unicamp, in 1993. From 1994 to 1996, he was with the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA, where he was involved in interactive image segmentation for the Ph.D. degree. He received the Ph.D. degree in electrical engineering from Unicamp, in 1996. In 1997, he worked on a project for Globo TV with the CPqD-TELEBRAS Research Center, Campinas, developing methods for video quality assessment. He was a Professor of Computer Science and Engineering with Unicamp in 1998.

His main research interests include image/video processing, visualization, and analysis, graph algorithms and dynamic programming, image annotation, organization, and retrieval, machine learning and pattern recognition, and image analysis applications in biology, medicine, biometrics, geology, and agriculture.



Anderson Rocha received the B.Sc. degree in computer science from the Federal University of Lavras, Brazil, in 2003, and the M.S. and Ph.D. degrees from the University of Campinas (Unicamp), Campinas, Brazil, in 2006 and 2009, respectively. He is currently an Assistant Professor with the Institute of Computing, Unicamp.

His main interests include digital forensics, reasoning for complex data, and machine intelligence. He has actively worked as a Program Committee Member in several important Computer Vision, Pattern Recognition, and Digital Forensics events. He is an Associate Editor of the *Elsevier Journal of Visual Communication and Image Representation*, and a Leading Guest Editor of the *EURASIP/Springer Journal on Image and Video Processing* and the *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*. He is an affiliate member of the Brazilian Academy of Sciences and the Brazilian Academy of Forensics Sciences. He is also a Microsoft Research Faculty Fellow of the IEEE Information Forensics and Security Technical Committee (IFS TC), and was a member of IFS TC.