Face Spoofing Detection on Low-Power Devices Using Embeddings with Spatial and Frequency-Based Descriptors

Rafael Henrique Vareto(✉), Matheus A. Diniz, and William Robson Schwartz

Smart Sense Laboratory, Department of Computer Science,
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
{rafaelvareto,matheusad,william}@dcc.ufmg.br

Abstract. A face spoofing attack occurs when an intruder attempts to impersonate someone with a desirable authentication clearance. To detect such intrusions, many researchers have dedicated their efforts to study visual liveness detection as the primary indicator to block spoofing violations. In this work, we contemplate low-power devices through the combination of Fourier transforms, different classification methods, and low-level feature descriptors to estimate whether probe samples correspond to spoofing attacks. The proposed method has low-computational cost and, to the best of our knowledge, this is the first approach associating features extracted from both spatial and frequency domains. We conduct experiments with embeddings of Support Vector Machines and Partial Least Squares on recent and well-known datasets under same and cross-database settings. Results show that, even though devised towards resource-limited single-board computers, our approach is able to achieve significant results, outperforming state-of-the-art methods.

Keywords: Face spoofing · Liveness detection · Fourier transform · Machine learning · Biometrics

1 Introduction

Biometric techniques seek for recognizing humans taking into account their intrinsic behavioral or observable aspects, ranging from face and fingerprint to iris and voice. Even though the biometric authentication field has prospered significantly in the recent years, experts claim that new technologies are constantly susceptible to malicious attacks and can be exposed to emerging high-quality spoof mechanisms [18].

Spoofing, also known as copy or presentation attack, is a real threat for biometric systems. More precisely, it occurs when an intruder attempts to impersonate someone who holds a desirable authentication clearance. The criminal usually employs falsified data to bypass the security procedure and gain illegitimate access. As a countermeasure to copy attacks, some researchers dedicate

I. Nyström et al. (Eds.): CIARP 2019, LNCS 11896, pp. 187–197, 2019.
https://doi.org/10.1007/978-3-030-33904-3_17

their efforts to study human liveness detection as the leading indicator to anticipate spoofing violations [10,15,16,19,28].

In general, a spoofing attack involves the display of still or motion pictures of authentic users registered in a set of known individuals present in a face recognition system. These images are easily acquired since the person's face is probably the most typical biometric model due to its noninvasive and availability characteristics when compared to others, such as fingerprint and iris. With the expansion of surveillance cameras and the increasing number of people distributing personal pictures on social networks, it is practically impossible to keep faces from spreading out [12]. Thus, face spoofing has become an easy approach to deceive biometric-based applications.

This paper is inspired on the works of Pinto et al. [20] and Vareto et al. [26]. However, due to the high demand for low computational-cost algorithms to be embedded on low power devices (e.g., IoT devices), we devise an anti-spoofing algorithm for limited-resource equipments. We propose a spoofing detection approach that associates simple handcrafted features extracted from spatial and frequency domains. Classifiers act as bootstrap aggregating meta-algorithms to achieve competitive results on the five most prominent benchmarks, to mention a few, MSU-MFSD [27], OULU-NPU [5] and SIW [14] datasets. We conduct cross-dataset experiments in the interest of assessing the method's generalization and verify how it responds to "unfamiliar" media presentations. This work compares the proposed method with state-of-the-art approaches and investigates how much display devices and image capture quality have an impact on our results.

To the best of our knowledge, this is the first approach associating features extracted from the spatial and frequency domains to tackle the spoofing detection problem. The leading premise is that modeling the association between spatial and frequency domains can be suitable for improving the accuracy and robustness of face anti-spoofing tasks. We assume that authentic and counterfeit biometric data enclose distinct noise signatures derived from the media acquisition. In fact, we believe that the combination of different feature descriptors contributes to achieving higher performance considering that they acquire distinctive characteristics, which are capable of enriching the classifier's robustness and generalization potential.

The main contributions of this work are: *(1)* combination of classification models fitted on randomly generated subsets in a bootstrap aggregating mode; *(2)* aggregation of features extracted in spatial and temporal domains; *(3)* efficient method for image and video-based copy attack receiving as input high-resolution videos; *(4)* low complexity and computational cost algorithm, capable of being deployed in embedded systems and computers with small processing capabilities; *(5)* clear study and experimental evaluation of the proposed approach considering fundamental feature descriptors, such as GLCM [11], HOG [8] and LBP [17].

2 Related Works

In the past years, Deep Neural Networks (DNN) have confirmed to be effective in several computer vision and biometric problems. Feng et al. [9] extract deep features from a convolutional neural network to identify real and fake faces. Similarly, Li et al. [13] employ a multiple-input hierarchical neural network combining either *shearlet* or optical-flow-based features. Valle et al. [25] present a transfer learning method using a pre-trained DNN model on static features to recognize photo, video and mask attacks. Liu et al. [14] combine DNN and Recurrent Neural Networks (RNN) to estimate the depth of face images along with rPPG signals to boost the detection of unauthorized access.

Some authors carry on working on long-established traditional approaches, dealing with handcrafted feature extraction and learning design: Pinto et al. [20] explore the spatial domain during the recapture process as it takes over the noise with Fourier transforms followed by visual rhythm algorithms and the extraction of gray-level co-occurrence matrices. Wen et al. [27] come up with an algorithm built on image distortion analysis and low-level feature descriptors. It consists of an embedding of SVM classification algorithms evaluated on cross-dataset scenarios. Pinto et al. [19] extract low-level feature descriptors gathering temporal and spectral information across biometric samples. Boulkenafet et al. [3,4] detect copy attacks using color texture analysis and low-level descriptors via exploring luminance and chrominance information of each image color channel separately.

Even though handcrafted features may end up being restricted to specific datasets domains, they are commonly faster and present lower memory usage than DNN-based methods, especially when it comes to resource-limited equipments. Most neural networks are not invariant to image rotation or scale and may fail to manage scenarios consisting of differing capturing instruments, illumination conditions and shooting angles [2]. In addition, top performing DNNs tend to suffer from either low speed or being too large to fit into single-board computers, preventing their deployment on remote applications. On the contrary of deep neural networks, both traditional features and straightforward classifiers employed in our approach do not require cloud processing services or powerful dedicated servers since embedded devices are capable of running the proposed low-cost standalone algorithm fast enough to be employed in real environments.

3 Proposed Approach

We propose an approach that captures visual noise signatures in both spatial and frequency domains. First, the method extracts low-level features with GLCM [11], HOG [8] and LBP [17]. Then, an ensemble of classifiers is created as we group several identical classifiers to enhance the method's overall efficacy [6]. Figure 1 illustrates the steps that compose the proposed approach.

Different feature descriptors make it possible to combine color, gradient magnitude and texture information, providing complementary evidence for presentation attacks. More precisely, GLCM is a statistical descriptor that analyses spatial

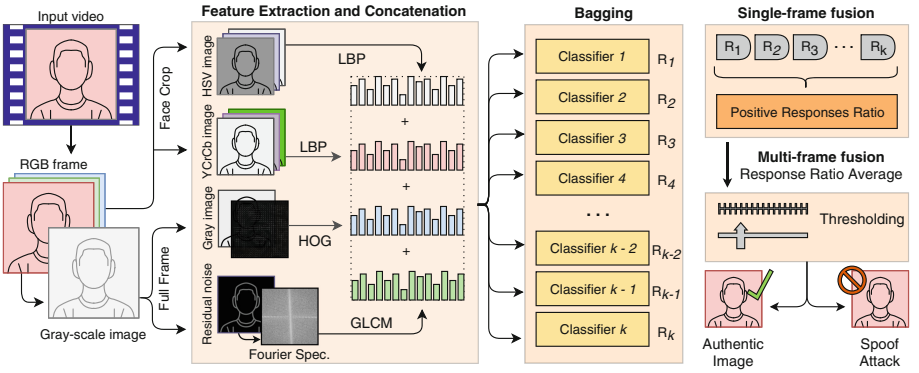

Fig. 1. Overview of the proposed face spoofing detection approach – *Training:* GLCM, HOG and LBP descriptors are extracted from the frames of the videos available for training. These features are concatenated and used for learning several classification models in an embedding fashion. Distinct models are learned containing different video samples in each subset. *Test:* The same features are extracted from the probe video frames and projected to all binary classifiers. Then, it executes a score fusion on the classifiers' responses to determine whether the probe video refers to an authentic presentation.

relationship of pixels and may identify noise artifacts originated from the recapturing process. HOG captures regions of abrupt intensity changes around edges and corners, such as screen frames and picture borders, through the magnitude of gradients. LBP evaluates color and texture patterns in search of *crude* attacks as it compares pixels with their surrounding points in different colorspaces.

3.1 Feature Extraction

The feature extraction process explores distinct spatial colorspaces and frequency domain to gather discriminating spoofing patterns. The procedure starts converting every RGB colorspace video frame into HSV, YC_RC_B and gray-scale images. On the contrary of the RGB color model, which holds high correlation among color components, HSV and YC_RC_B are capable of isolating luminance from chrominance and more robust to illumination variations [21].

As the RGB video frame is converted into HSV and YC_RC_B images, the method locates the region of interest, which is delimited on the subject's face. The approach extracts LBP descriptors from each HSV and YC_RC_B image color channel in an attempt to gather color and texture distinctive information. In fact, it computes local texture representation from all color bands comparing every pixel with its surrounding neighborhood of pixels. Both HSV and YC_RC_B corresponding feature descriptors derive from the integration of each channel's histogram that accounts for the number of times every LBP pattern occurs [4].

Monochromatic video frames go through low-pass filtering techniques (blurring) for artifact and noise reduction. Residual noises are then obtained by subtracting a gray-scale image and its slightly blurred version [20]. A logarithmic-

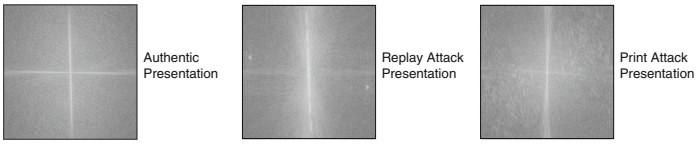

Fig. 2. Comparison among Fourier spectra extracted from different presentation images. Note that there are some artifacts spread throughout print and replay attacks.

scaled Fourier transform function $\mathcal{F}_{log}(v,u)$ decomposes each residual image $r(x,y)$ of size $M \times N$ into its sine and cosine components where each pixel constitutes a frequency from the spatial domain as

$$\mathcal{F}_{log}(v,u) = log(1 + |\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r(x,y)e^{-j2\pi[\frac{vx}{M}+\frac{uy}{N}]}|).$$

The employed low-level feature descriptors provide great accuracy vs. speed trade-off due to their fast computation. The gray-scale image and its corresponding spectrum generate HOG and GLCM features, respectively, whereas LBP descriptor receives HSV and $\mathrm{YC_RC_B}$ image color bands. HOG carries shape information by counting occurrences of gradient orientation using histograms while GLCM measures the residual image texture with the generation of co-occurring gray-scale values at a determined offset. As shown in Fig. 1, we concatenate HOG and LBP features from the spatial-domain with GLCM information from the *log*-scaled Fourier spectrum to build a robust feature descriptor.

3.2 Classification Methods

Instead of learning a unique binary classifier, we learn a set of models as it seems to be more appropriate to handle contrasting chromatic distortions and to reduce the risk of overfitting. The classification embedding consists either of Support Vector Machines (SVM) [24] or Partial Least Squares (PLS) [22] learning algorithms. While the former chooses the hyperplane that maximizes the distance to the nearest data points, the latter weights features to discriminate throughout different classes and handle high-dimensional data.

During the training stage, the proposed method employs several identical binary learning algorithms trained on random subsets of the training set to create an array of classifiers C. It guarantees a balanced division within each classification model since v genuine live and v presentation attack videos are randomly selected, with replacement, out of all video samples available for training. Then, it fits the learning algorithm on the extracted features where the positive class only contains "authentic" feature vectors and the negative class holds features extracted from copy attacks. This process is repeated k times, where $k = |C|$ is a user-defined parameter that defines the number of classification models.

In the prediction stage, the method projects every single frame onto all classification models as it iterates over the probe video. For each frame, the algorithm

computes the ratio of the number of positive responses attained to the total number of classification models k. If most $c \in C$ classifiers return positive responses, it implies that the frame is likely to be a *bona fide* (authentic) sample. Otherwise, if they return negative responses, then the probe sample is likely to belong to a spoofing attack. As the approach examines multiple frames of a probe video, it obtains the numerical mean of all frame ratio scores. A probe video is considered authentic if the averaged ratio score of all frames satisfies a threshold t (t would be chosen according to the biometric system specifications).

4 Experimental Results

This section contains an objective evaluation of the proposed algorithm, which generates many binary classification models combined with a majority voting scheme that determines whether a query image corresponds to a legitimate image or a spoofing attack.

Table 1. Evaluation on different SIW protocols with an increasing number of PLS classification models (PLS approach). Note that the method becomes more discriminative with the addition of classifiers.

Protocol	Metric	50	100	200
1	APCER	0.68 ± 0.00	0.14 ± 2.17	0.00 ± 0.00
	BPCER	4.67 ± 0.00	2.17 ± 0.00	0.67 ± 0.00
2	APCER	11.70 ± 10.73	7.86 ± 6.84	3.93 ± 4.14
	BPCER	3.34 ± 4.74	1.29 ± 1.16	0.66 ± 1.10
3	APCER	17.37 ± 14.53	10.59 ± 7.72	6.99 ± 1.68
	BPCER	4.92 ± 4.09	2.17 ± 1.67	1.17 ± 0.33

Feature Descriptors. Three feature descriptors are employed in this work: The GLCM texture descriptor [11] is computed with directions $\theta \in \{0, 45, 90, 135\}$ degrees, distance $d \in \{1, 2\}$, 16 bins and six texture properties: contrast, dissimilarity, homogeneity, energy, correlation, and angular second moment. The HOG shape descriptor [8] is set with 96×96 cells and holding eight orientations. Lastly, the LBP texture descriptor [17] comprises 256 bins, a radius equal to 1, and eight points arranged in a 3×3 matrix thresholded by its central point. Their low complexity and computational cost endorse our method so that it can be deployed to embedded systems with reduced processing capabilities.

Spoofing Datasets. For a thorough evaluation, we select datasets with distinct protocols, medium characteristics and different lighting conditions. Therefore, experiments are carried out on five benchmarks: CASIA-FASD [29], MSU-MFSD [27], OULU-NPU [5], REPLAY-ATTACK [7] and SIW [14]. CASIA-FASD, MSU-MFSD and REPLAY-ATTACK are traditional benchmark databases made up of genuine live

recordings and distinct spoofing attack shots captured by distinct cameras in different scenarios. Both OULU-NPU and SIW are recent datasets containing full high-definition videos of multiethnic individuals and featuring 30-FPS live and presentation attack videos.

Evaluation Metrics. We employ ISO/IEC 30107-3 metrics [1] called Attack Presentation Classification Error Rate, APCER $= \frac{1}{V_{PA}} \sum_{i=1}^{V_{PA}} (1 - Res_i)$; and Bona Fide Presentation Classification Error Rate, BPCER $= \frac{1}{V_{BF}} \sum_{i=1}^{V_{BF}} (Res_i)$. V_{PA} indicates spoofing attacks whereas V_{BF} outlines authentic presentations. Res_i receives 0 when the i-th probe video is considered an *bona fide* presentation and 1 otherwise. On cross-datasets evaluations, it is customary to employ Half Total Error Rate, HTER $= \frac{\text{FAR}+\text{FRR}}{2}$, which is half the sum of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) [14,23]. The reader must bear in mind that the closer APCER, BPCER and HTER values get to zero, the more accurate the described methods are.

Evaluation Setup. Experiments were conducted on a Raspberry Pi 3 Model B and on a Linux virtual machine to assess the performance of the proposed approach on different machines. First, we analyzed the method on a CPU-based machine consisting of eight 2.0 GHZ-core processors and 16 GB RAM memory, but no more than 600 MB was required on test time. Then, we migrated to the Raspberry, a single-board microcomputer with a 1.2 GHZ Quad Core CPU and 1 GB RAM memory. Higher frame rates could be achieved with graphical processing units, but it would demand the acquisition of more advanced hardware.

Table 2. APCER and BPCER results (%) on SIW protocols.

Protocol	Method	APCER	BPCER	AVERAGE
1	Deep models [14]	3.58	3.58	3.58
	PLS approach	0.00 ± 0.00	0.67 ± 0.00	0.33 ± 0.00
	SVM approach	0.00 ± 0.00	0.33 ± 0.00	0.16 ± 0.00
2	Deep models [14]	0.57 ± 0.69	0.57 ± 0.69	0.57 ± 0.69
	PLS approach	3.93 ± 4.14	0.66 ± 1.10	2.24 ± 3.30
	SVM approach	8.09 ± 1.02	1.00 ± 0.44	2.88 ± 3.21
3	Deep models [14]	8.31 ± 3.81	8.31 ± 3.80	8.31 ± 3.81
	PLS approach	6.99 ± 1.68	1.16 ± 0.33	1.55 ± 0.05
	SVM approach	6.03 ± 1.22	0.67 ± 0.24	3.36 ± 0.73

Results Analysis. The algorithm proposed in Sect. 3 is evaluated according to the protocols available in the literature and following the datasets instructions. For databases containing only training and test sets, like SIW dataset, we reserve ten percent of all samples available for training to establish an automatic adaptive threshold t. Differently, OULU-NPU and REPLAY-ATTACK contain a development set destined to parameter calibrations.

Table 3. APCER and BPCER results (%) on OULU-NPU protocols.

Protocol	Method	APCER	BPCER	AVERAGE
1	Deep models [14]	1.60	1.60	1.60
	Gradiant [2]	1.30	12.50	6.90
	PLS approach	5.50 ± 2.11	9.79 ± 3.37	7.64 ± 2.74
2	Deep models [14]	2.70	2.70	2.70
	Gradiant [2]	6.90	2.50	4.70
	PLS approach	2.13 ± 1.07	3.61 ± 1.21	2.87 ± 1.14
3	Deep models [14]	2.70 ± 1.30	3.10 ± 1.70	2.90 ± 1.50
	Gradiant [2]	2.60 ± 3.90	5.00 ± 5.30	3.80 ± 2.40
	PLS approach	3.12 ± 2.58	8.51 ± 6.20	5.81 ± 4.39
4	Deep models [14]	9.31 ± 5.60	10.4 ± 6.00	9.50 ± 6.00
	Gradiant [2]	5.00 ± 4.50	15.0 ± 7.10	10.0 ± 5.01
	PLS approach	17.8 ± 9.83	9.37 ± 4.31	13.5 ± 7.07

We evaluate the method's behavior by increasing the number of PLS classification models. According to the results showed in Table 1, as the number of classifiers increases, the method becomes more discriminative. Therefore, in the remaining experiments, we set the number of classification models to 200. Tables 2 and 3 show the results obtained on the SIW and OULU-NPU datasets, respectively. The proposed approach achieves state-of-the-art results on SIW Protocols 1 and 3 and competitive results on Protocol 2. Moreover, the method attains precise results on three out of four OULU-NPU Protocols.

The cross-database analysis provides an insight into countermeasure methods' generalization power. In this sort of scenario, an algorithm is trained and tuned in one of the datasets and tested on the others. Table 4 presents the cross-testing HTER [1] performance for both PLS and SVM methods on the traditional benchmarks. The PLS-based method also achieves a HTER of 34.44 ± 3.91 when trained on SIW and tested on OULU-NPU, and 17.55 ± 1.47 vice versa. Results show that datasets tend to hold some bias regardless of their protocols due to the intrinsic and specific information enclosed in each dataset, culminating in a significant accuracy reduction when compared to same-database evaluations.

Computational Cost Evaluation. In constrast to most recent spoofing detection works in the literature, where deep neural networks benefit from "unlimited computational resources" and high-bandwidth video transmissions, our method is devised towards resource-limited single-board computers in order to reduce network communication. GLCM, HOG and LBP descriptors appear to carry relevant forensic signature information of image and video-based spoofing detection since results show that the combination of spatial and frequency-based descriptors contributes to achieving both competitive and state-of-the art results.

Table 4. Cross-dataset evaluation (%) presenting HTER metric on CASIA-FASD, MSU-MFSD and REPLAY-ATTACK datasets.

Training Set	CASIA-FASD		MSU-MFSD		REPLAY-ATTACK	
Test Set	MSU-MFSD	REPLAY-ATTACK	CASIA-FASD	REPLAY-ATTACK	CASIA-FASD	MSU-MFSD
Color LBP [3]	36.6	47.0	49.6	42.0	39.6	35.2
Color texture [4]	20.4	30.3	46.0	33.9	37.7	34.1
Spectral [19]	-	34.4	-	-	50.0	-
Deep models [14]	-	27.6	-	-	28.4	-
PLS approach	19.2 ± 1.6	30.1 ± 0.7	28.2 ± 0.7	37.1 ± 3.2	35.6 ± 0.4	34.5 ± 2.3
SVM approach	17.3 ± 1.1	42.6 ± 2.5	34.8 ± 0.8	42.6 ± 1.7	38.3 ± 2.0	35.4 ± 1.9

Many researchers have neglected to deliver biometric applications that are able to run on low-power devices [9,13,14,25]. As we take IoT devices into account, the proposed algorithm presents low computational cost, being able to process up to 4.31 ± 0.031 frames per second (FPS) when considering the Raspberry Pi environment. As a comparison, it runs at 32.55 ± 0.96 FPS in the CPU-based computer. Both when the number of classifiers k is set to 100. Such frame rate, 4.31 FPS, make it feasible for tech developers to implement and run biometric IoT technologies in real environments.

When we consider the above frame rate specification and the average amount[1] paid for the following devices: a Raspberry Pi 3 Model B ($35.00), identical to the microcomputer evaluated; an Intel i5 2.8 GHZ processor with 16 GB RAM ($400.00), similar to the virtual machine tested; and an Intel i7 3.2 GHZ CPU with 16 GB RAM and a GeForce GTX 1080Ti ($1600.00), assuming an equivalent frame rate of 32.55, since most quality CCTV cameras record videos between 15 and 30 FPS. Then, the price paid per FPS on the aforementioned machines would be around $8.12, $12.28 and $49.15, respectively. Therefore, running the designed approach on a single-board computer, such as Raspberry Pi, provides better performance per cost than executing in more robust machines.

5 Conclusions

This work[2] proposed a fast and low-memory spoofing detection algorithm and demonstrates how it performs in an experimental setup to emulate real-world scenarios. The proposed algorithm is fast and works well on single-board computers with high-resolution videos and is able to achieve state-of-the-art performance on widely explored databases.

We conduct an objective investigation on how far spatial and frequency-based descriptors can get when combined with multiple classification models. If fact, we work out two approaches (embeddings comprised of either Partial Least Squares

[1] Prices taken from official Raspberry Pi resellers and BestBuy Retail Store.
[2] Proposed method available at https://github.com/rafaelvareto/Spoofing-CIARP19.

or Support Vector Machines) to infer that the association of long-established feature descriptors accomplish great performance in same-database settings. An investigation carried out on different datasets show that the accuracy tends to degrade significantly.

Despite the great progress in several biometric research areas, existing anti-spoofing approaches have shown lack of generalization in cross-dataset conditions, which best represents real-world scenarios. As future directions, we plan to add extra feature descriptors, include other relevant spoofing datasets and learn spatial-temporal representations.

Acknowledgments. The authors would like to thank the Brazilian National Research Council – CNPq (Grants #311053/2016-5 and #438629/2018-3), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17), the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project), Maxtrack Industrial LTDA and Empresa Brasileira de Pesquisa e Inovacao Industrial – EMBRAPII.

References

1. Information technology - biometric presentation attack detection - part 1: Framework. international organization for standardization. Technical report, ISO/IEC JTC 1/SC 37 Biometrics (2016)
2. Boulkenafet, Z., et al.: A competition on generalized software-based face presentation attack detection in mobile scenarios. In: IJCB, pp. 688–696. IEEE (2017)
3. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face anti-spoofing based on color texture analysis. In: ICIP, pp. 2636–2640. IEEE (2015)
4. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. TIFS **11**(8), 1818–1830 (2016)
5. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: OULU-NPU: a mobile face presentation attack database with real-world variations. In: FG, IEEE (2017)
6. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
7. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: BIOSIG. No. EPFL-CONF-192369 (2012)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893. IEEE (2005)
9. Feng, L.: Integration of image quality and motion cues for face anti-spoofing: a neural network approach. JVCIR **38**, 451–460 (2016)
10. Garcia, D.C., de Queiroz, R.L.: Face-spoofing 2D-detection based on moiré-pattern analysis. TIFS **10**(4), 778–786 (2015)
11. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. TSMC **6**, 610–621 (1973)
12. Kumar, S., Singh, S., Kumar, J.: A comparative study on face spoofing attacks. In: ICCCA, pp. 1104–1108. IEEE (2017)
13. Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A.: An original face anti-spoofing approach using partial convolutional neural network. In: IPTA, pp. 1–6. IEEE (2016)
14. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In: CVPR, pp. 389–398 (2018)

15. Määttä, J., Hadid, A., Pietikäinen, M.: Face spoofing detection from single images using micro-texture analysis. In: IJCB, pp. 1–7. IEEE (2011)
16. Menotti, D., et al.: Deep representations for iris, face, and fingerprint spoofing detection. TIFS **10**(4), 864–879 (2015)
17. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI **24**(7), 971–987 (2002)
18. Pereira, T., Anjos, A., Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: ICB, pp. 1–8. IEEE (2013)
19. Pinto, A., Pedrini, H., Schwartz, W.R., Rocha, A.: Face spoofing detection through visual codebooks of spectral temporal cubes. TIP **24**(12), 4726–4740 (2015)
20. Pinto, A., Schwartz, W.R., Pedrini, H., de Rezende Rocha, A.: Using visual rhythms for detecting video-based facial spoof attacks. TIFS **10**(5), 1025–1038 (2015)
21. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-662-04186-4
22. Rosipal, R., Krämer, N.: Overview and recent advances in partial least squares. In: Saunders, C., Grobelnik, M., Gunn, S., Shawe-Taylor, J. (eds.) SLSFS 2005. LNCS, vol. 3940, pp. 34–51. Springer, Heidelberg (2006). https://doi.org/10.1007/11752790_2
23. Siddiqui, T.A., et al.: Face anti-spoofing with multifeature videolet aggregation. In: ICPR, pp. 1035–1040. IEEE (2016)
24. Steinwart, I., Christmann, A.: Support Vector Machines. Springer, Heidelberg (2008). https://doi.org/10.1007/978-0-387-77242-4
25. Lucena, O., Junior, A., Moia, V., Souza, R., Valle, E., Lotufo, R.: Transfer learning using convolutional neural networks for face anti-spoofing. In: Karray, F., Campilho, A., Cheriet, F. (eds.) ICIAR 2017. LNCS, vol. 10317, pp. 27–34. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59876-5_4
26. Vareto, R., Silva, S., Costa, F., Schwartz, W.R.: Towards open-set face recognition using hashing functions. In: IJCB, pp. 634–641. IEEE (2017)
27. Wen, D., Han, H., Jain, A.K.: Face spoof detection with image distortion analysis. TIFS **10**(4), 746–761 (2015)
28. Xiong, Q., Liang, Y.C., Li, K.H., Gong, Y.: An energy-ratio-based approach for detecting pilot spoofing attack in multiple-antenna systems. TIFS **10**(5), 932–940 (2015)
29. Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z.: A face antispoofing database with diverse attacks. In: ICB, pp. 26–31. IEEE (2012)