Deep Texture Features for Robust Face Spoofing Detection

Gustavo Botelho de Souza, *Student Member, IEEE*, Daniel Felipe da Silva Santos, Rafael Gonçalves Pires, Aparecido Nilceu Marana, and João Paulo Papa, *Member, IEEE*

***Abstract*—Biometric systems are quite common in our everyday life. Despite the higher difficulty to circumvent them, nowadays criminals are developing techniques to accurately simulate physical, physiological, and behavioral traits of valid users, process known as spoofing attack. In this context, robust countermeasure methods must be developed and integrated with the traditional biometric applications in order to prevent such frauds. Despite face being a promising trait due to its convenience and acceptability, face recognition systems can be easily fooled with common printed photographs. Most of state-of-the-art antispoofing techniques for face recognition applications extract handcrafted texture features from images, mainly based on the efficient local binary patterns (LBP) descriptor, to characterize them. However, recent results indicate that high-level (deep) features are more robust for such complex tasks. In this brief, a novel approach for face spoofing detection that extracts deep texture features from images by integrating the LBP descriptor to a modified convolutional neural network is proposed. Experiments on the NUAA spoofing database indicate that such deep neural network (called LBPnet) and an extended version of it (n-LBPnet) outperform other state-of-the-art techniques, presenting great results in terms of attack detection.**

***Index Terms*—Face recognition, spoofing detection, biometrics, deep texture features, convolutional neural networks.**

I. Introduction

BIOMETRICS, i.e., people automatic recognition based on their physical, physiological or behavioral traits, presents many advantages over the traditional knowledge- and possess-based identification systems [1], [2]. However, nowadays criminals are already developing sophisticated techniques to accurately simulate traits of valid users, process known as spoofing attack. In this sense, countermeasures, i.e., robust spoofing detection techniques, must be developed and integrated with biometric systems in order to prevent such frauds [3].

There are many points of attack in security systems that can be exploited by criminals. In the case of biometric systems, the great majority of attacks occur by fooling the capture sensor with synthetic traits since no knowledge regarding the inner working of the application is needed [4]. Among the main biometric traits, face is a promising one especially due to its convenience, low cost of acquisition and acceptability by users, being very suitable to a wide variety of environments, including mobile ones. However, despite all these advantages, face recognition systems are the ones that most suffer with spoofing attacks since they can be easily fooled even with common printed photographs obtained in the worldwide network.

In this brief a novel approach for face spoofing detection that works with high-level (deep) texture features instead of handcrafted ones is proposed based on a modified Convolutional Neural Network (CNN) [5] by incorporating the LBP (Local Binary Patterns) [6] texture descriptor in its first layer. Besides the good results of LBP itself, CNNs have been increasingly used in such difficult tasks since they can extract and work accurately with high-level features, learned from the own set of training data, being more robust and suitable for activities such as attack detection, in which patterns are complex and can not be easily detected. Experiments show that the proposed deep neural network, called LBPnet, and its extended version, n-LBPnet (normalized LBPNet), outperform the state-of-the-art techniques based on handcrafted texture information, presenting great results in terms of attack detection.

II. Texture-Based Face Spoofing Detection

Likewise as in face recognition, texture plays an important role in face spoofing detection [7]. In general, the state-of-the-art antispoofing techniques extract handcrafted texture features from images in order to detect fake faces [7]–[10]. Among the main texture descriptors, the original LBP (Local Binary Patterns) [6] and its variations are the most commonly used due the good results they allow to reach and the efficient algorithm of LBP.

Briefly explaining, given a neighborhood system $\{P, R\}$, where P corresponds to the number of neighbors to be considered and R the radius of the neighborhood, the LBP descriptor works by comparing the grayscale value of each pixel p of the image with the intensity of its neighbors and associating a new integer (grayscale) value to p based on such comparisons.

Manuscript received May 12, 2017; revised August 31, 2017; accepted October 7, 2017. Date of publication October 20, 2017; date of current version November 22, 2017. This work was supported in part by the São Paulo Research Foundation under Grant 2014/12236-1 and Grant 2017/05522-6, in part by the National Council for Scientific and Technological Development under Grant 306166/2014-3, and in part by the Brazilian Coordination for the Improvement of Higher Education Personnel under Grant 88881.132647/2016-01. This brief was recommended by Associate Editor C. W. Wu. *(Corresponding author: Aparecido Nilceu Marana.)*

G. B. de Souza and R. G. Pires are with the CCET—Exact and Technology Sciences Center, Federal University of São Carlos, São Carlos 17060-326, Brazil (e-mail: gustavo.botelho@gmail.com; rafapires@gmail.com).

D. F. da Silva Santos, A. N. Marana, and J. P. Papa are with the Department of Computing, Faculty of Sciences, UNESP—São Paulo State University, São Carlos 17033-360, Brazil (e-mail: danielfssantos1@gmail.com; nilceu@fc.unesp.br; papa@fc.unesp.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/TCSII.2017.2764460

In handcrafted methods, in general, a histogram is built based on the grayscale values of the LBP-based image in order to represent its original version and, given a set of known images and their respective histograms, a classifier, e.g., a Support Vector Machine (SVM) [11], is trained to predict the classes of new faces (real or fake) [7]. Some works divide the face image in patches and generate a histogram to each patch, which at the end are concatenated.

An important handcrafted method for face spoofing detection is based on a multiscale version of LBP (Multiscale Local Binary Patterns - MLBP) [7]. It generates multiple handcrafted histograms from a given image, varying the neighborhood (values of P and R) of the LBP descriptor, to characterize it. Other recently proposed technique, e.g., is based on Dynamic Local Ternary Patterns (DLTP) [8]. It uses a modified version of LBP, which works with three labels instead of two when comparing the central pixel with its neighbors, to classify faces in real or synthetic ones.

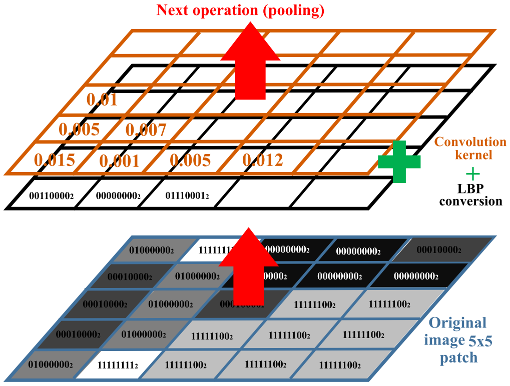

Fig. 1. The convolution operation in the first layer of the proposed LBPnet actuates converting the image to its LBP-based version (binary values are shown in black just to clarifying, these values are converted to the respective integer - grayscale - ones) and convolving it with predefined kernels (example of values in orange).

III. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) [5] are deep learning architectures constituted of layers in which different kind of filters (convolution and sampling) are applied to the input data, initially two-dimensional images. The result of a given layer serves as input to the above one until the top of the network is reached. Differently from the fully connected networks, CNNs present a simplified topology and neurons of same layers use to share parameters, enabling an efficient learning. Besides convolutional and sampling operations, layers with neurons completely connected can be included at the top of the network for classification [12]–[14].

In practice, given a two-dimensional image, in each network layer a set of convolutional filters (kernels) are applied, obtaining different channels of the original input. Pooling, i.e., sampling operations are also performed in order to obtain certain kind of translational and scale invariance and reduce the amount of data being considered. At the top of the network it is obtained a high-level representation of the original image, which is more robust than the raw pixels information for many applications [3].

IV. LBP-Based Convolutional Neural Network

Based on the well-referenced Lenet-5 [5] network model, in this brief a novel CNN architecture, called LBPnet, is proposed by integrating the LBP (Local Binary Patterns) [6] descriptor in its first layer in order to extract deep texture features, instead of handcrafted histograms, from images for a more robust face spoofing detection.

The first layer of LBPnet incorporates LBP information as follows: the convolution operation actuates not only convolving the values of the kernels (weights of connections between neurons learned in training) with the image grayscale values, but also finding the LBP values of the image pixels before performing the convolution, i.e., the convolution is performed on the transformed LBP values of the pixels and not on their original grayscale values. This improves a lot the results of the proposed deep neural network since the method inherits the power of enhancing face spoofing cues from the LBP descriptor in a deeper and more robust architecture, working with high-level texture features learned from the training data. Fig. 1 shows the convolution of the first layer of LBPnet.

The LBPnet presents the following configuration, from bottom to top, mainly inherited from Lenet-5: (i) Two layers with a convolution followed by a pooling operation - the first layer is modified, as said, by incorporating the LBP descriptor in the convolution step; (ii) a Rectified Linear Unit (ReLU) layer, that performs an inner product followed by a rectification (elimination of negative values) on the originated signals; and (iii) a Fully Connected (FC) layer, with two nodes, which also performs an inner product and classification (attack or not attack attempt) of the input image using the softmax function.

A scheme of the architecture of LBPnet is shown in Fig. 2. Given a detected and normalized grayscale facial image (in this brief resized to 66 × 66), the convolution operation in the first layer, *CONV1*, finds the pixels LBP-based values and produces 20 outputs with size 60 × 60 by convolving such values with 20 different kernels with size of 5 × 5 - each kernel generates an output and is applied with stride of 1 to the image.

The convolution operation can be written as:

$$C_i(p) = \sum_{\forall q \in N(p)} LBP(I(q)) \cdot K_i(j) \tag{1}$$

where $LBP(I(q))$, with $q = (x_q, y_q)$, represents the LBP value of the pixel q belonging to the neighborhood of pixel $p = (x_p, y_p)$, i.e., $q \in N(p)$ in original image I (also considering $p \in N(p)$); $C_i(p)$ means the value in the corresponding position to p in the output feature map C_i, with $i = 1, 2, \ldots, 20$; and $K_i(j)$, also with $i = 1, 2, \ldots, 20$, corresponds to the value in the i^{th} convolution kernel in the respective position of q.

Actually, the values of the kernels, $K_i(j)$, with $i = 1, 2, \ldots, 20$ and $j = (1, 1), (1, 2), \ldots, (5, 5)$, are generally initialized with weights inversely proportional to the size of input and output matrices of the convolution operation and also can be viewed as the weights of connections between the neurons

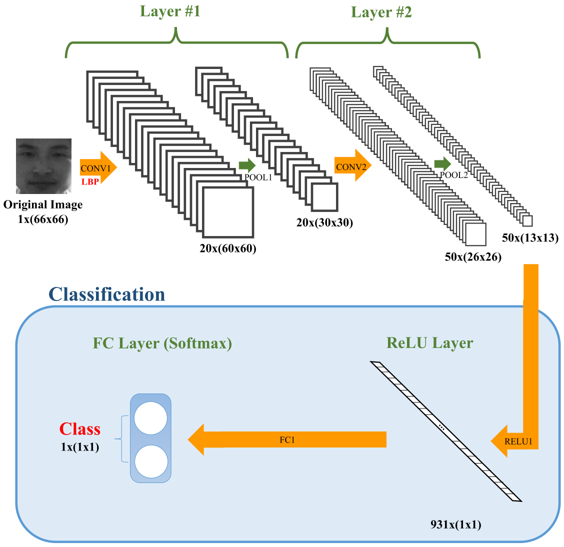

Fig. 2. Architecture of LBPnet. The first two layers perform convolution and pooling operations, (*CONV1*, *POOL1*) and (*CONV2*, *POOL2*), respectively. *CONV1* actuates not only convolving the input data with predefined kernels but also converting it to its LBP-based version before convolution.

of the CNN, which are adjusted during learning. The biases of such neurons are initially set to zero and also are optimized during training.

Still in the first layer of the LBPnet, a pooling operation, *POOL1*, is applied to obtain certain scale and translational invariance. In such case, the max-pooling is performed with a 2×2 sized kernel with no overlapping (stride of 2) generating 20 output feature maps with size 30×30 (since the size of the pooling kernel is 2 and there is no overlapping, the dimensions of the output feature maps of the pooling operation are half of the dimensions of the input ones). The max-pooling operation can be written as:

$$P_i(r) = max\{C_i(s)\}_{\forall s \in N(r)} \tag{2}$$

where $C_i(s)$, with $s = (x_s, y_s)$, represents the value in position s belonging to the neighborhood of position $r = (x_r, y_r)$, i.e., $s \in N(r)$, in feature map C_i (generated in the previous convolution step), with $i = 1, 2, \ldots, 20$ (also considering $r \in N(r)$); $P_i(r)$ means the value in the corresponding position to r in the new output feature map P_i, also with $i = 1, 2, \ldots, 20$.

These two mentioned operations, convolution and pooling, are repeated in the second layer of LBPnet, without LBP calculation, but also using kernels with size and stride of 5 and 1, and of 2 and 2, respectively. As shown in Fig. 2, after the second layer, there are 50 two-dimensional feature maps with size 13×13. At the top of the network there are a Rectified Linear Unit (ReLU) and a Fully Connected (FC) layers. The ReLU layer actuates by performing an inner product with the 13×13 structures and by rectifying the signal obtained, not propagating negative values, following Eq. 3:

$$ReLU(t) = max\{0; t\} \tag{3}$$

where t corresponds to the weighted sum of the signals from the neurons of the previous layer of the LBPnet (values in the 13×13 feature maps).

The Fully Connected layer presents two neurons fully connected to the neurons of the ReLU layer also performing an inner product operation and applying the softmax function for defining their activations, which is given by:

$$s_k = \frac{e^{u_k}}{e^{u_0} + e^{u_1}} \tag{4}$$

where u_k, with $k \in \{0; 1\}$, corresponds to the weighted sum of the values provenient from the previous layer (ReLU) in softmax neuron k. The softmax function normalizes the sum of inputs of both neurons, making that the sum of their outputs becomes 1 (probabilities of activation). It is used to classify the original image as spoofing (greater activation of the first neuron) or not (greater activation of the second neuron).

An extended version of LBPnet, called normalized LBPnet (n-LBPnet), is also proposed in this brief. The n-LBPnet architecture is quite similar to the LBPnet model, however a normalization step is included in the second layer of the network between the convolution and the pooling operation. First, the output of *CONV2*, i.e., each value in the resultant feature maps C_i, with $i = 1, 2, \ldots, 50$, is linearly rectified as in the ReLU layer of the network (following Eq. 3). After, each position $p = (x_p, y_p)$ in a given feature map C_i, i.e., $C_i(p)$ is normalized by considering the values in the same positions of the two previous, $C_{i-2}(p)$ and $C_{i-1}(p)$, and of the two posterior feature maps, $C_{i+1}(p)$ and $C_{i+2}(p)$, following:

$$C_i(p) = \frac{C_i(p)}{(1 + \alpha \sum_{j=i-2}^{i+2} C_j(p)^2)^{\beta}} \tag{5}$$

where α and β are parameters that control the magnitude of the normalization; α is generally set to 0.2 since 5 adjacent feature maps are considered, in our case, per time, and β is usually set to 0.75 (we used such values in the experiments of Section V). When there are no previous or posterior feature maps for a given C_i, zero-valued feature maps are virtually included to the output of *CONV2* to avoid problems with missing neighbors.

Such normalization step, called Local Response Normalization (LRN) [15], is usually included between the convolution and pooling operations in CNNs and simulates the competitive process presented by neurons of human brain in nearby areas, in which some neurons tend to inhibit activations of their neighbors when they are highly excited, enhancing the especialization of such cells to input signals and improving learning. It is important to note that, depending on the case, j may vary for more than 5 feature maps per time.

V. Experiments, Results and Discussion

The proposed networks, LBPnet and n-LBPnet, were assessed on the traditional NUAA Photograph Imposter Database [16], with images obtained from real and fake faces. This dataset contains 3,491 images for training (1,743 from real faces and 1,748 from printed ones) and 9,123 test images (3,362 real and 5,761 fake facial images). They were obtained from different people in terms of gender, age, etc., and on different capture sessions (also varying the cameras used for such task), making the database very realistic. Some examples of the captured images for composing such dataset are shown in Fig. 3 together with their normalized versions (after applying face detection, grayscale conversion and resizing algorithms).

The normalized images (in grayscale and with size of 64×64) were already provided by the authors of the database

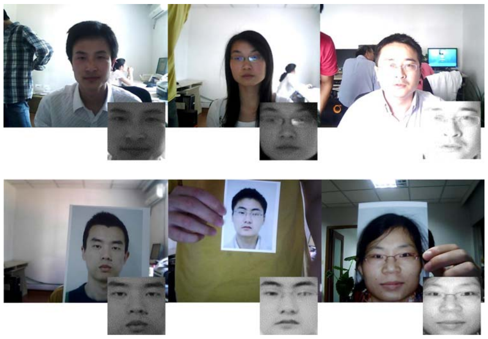

Fig. 3. NUAA [16] images with real (first row) and fake faces (second row). As one can observe, given the normalized grayscale images, attack detection is a challenging task.

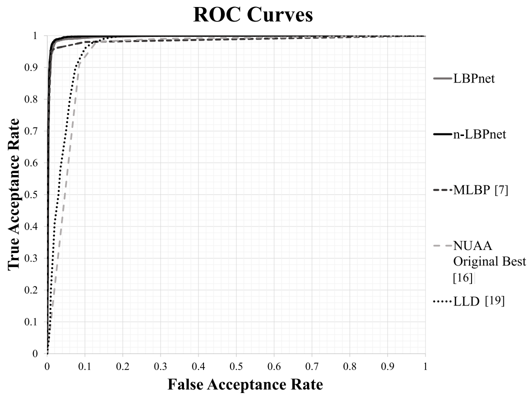

Fig. 4. ROC curves of the proposed methods, MLBP (Multiscale LBP) [7], the best method proposed by the authors of the NUAA [16] dataset, and of LLD (Low Level Descriptors) [19] approach. The higher the curve, the better the method.

in order to make the comparison of antispoofing methods fair, avoiding that different preprocessing techniques affect the results. We used such normalized images in our experiments. They were only resized to 66×66 pixels before feeding LBPnet and n-LBPnet since the LBP descriptor reduces the image dimensions by 2 pixels, going back to the size of 64×64 (we considered a neighborhood of $P = 8$ and $R = 1$ for LBP). As an observation, we augmented the training set (doubling its size) by considering the 3,491 initial normalized images and their histogram equalized versions in order to avoid lack of data while training the networks.

LBPnet and n-LBPnet were implemented using the Caffe [17] framework. As said, the weights of the kernels of both networks were initialized with values inversely proportional to the size of input and output feature maps in each operation ("xavier filler" from Caffe framework), while the biases of the neurons were zero-initialized. The networks were trained for 200 iterations by means of Stochastic Gradient Descent (SGD) [18] approach with the following parameters: 64 images per batch, initial learning rate of 0.01, momentum of 0.9 and weight decay of 0.004. The learning rate decay policy was "inv", i.e., the actual learning rate for a given training iteration k was given by:

$$lr_k = lr_0(1 + \gamma \cdot k)^{-\delta} \tag{6}$$

where lr_0 was the initial learning rate; γ was set to 0.0001 and δ to 0.75.

In order to evaluate the performances of LBPnet and n-LBPnet and compare with state-of-the-art methods, many metrics used in different works over the given NUAA [16] dataset were calculated, such as the ROC (Receiver Operating Characteristic) curve, Accuracy rate (Acc) and HTER (Half-Total Error Rate). Depending on the compared methods, a kind of metric was reported in literature, so we were not able to compared all of them with the proposed ones through all the metrics.

Regarding the ROC curves, Fig. 4 shows the True Acceptance Rate (TAR) *versus* the False Acceptance Rate (FAR) of: (i) n-LBPnet; (ii) LBPnet; (iii) the MLBP-based method [7]; (iv) the best method of the original paper of the NUAA [16] dataset - this best approach works on DoG (Difference of Gaussians) images with a sparse low rank bilinear logistic regression classifier; and (v) the Low Level Descriptors (LLD) [19] approach, i.e., combination of HoG (Histograms of Oriented Gradient), GLCM (Gray Level Co-occurrence Matrix) and HSC (Histograms of Shearlet Coefficients), which works with a Partial Least Squares (PLS) classifier. The higher the ROC curve, the better the approach. As can be seen, the proposed deep networks outperformed the best technique of the original paper of NUAA [16] database and the LLD approach [19] by far (both based on handcrafted texture features), presenting considerably higher curves.

Despite the MLBP-based approach [7] also presenting a high curve, it is still lower then the results of LBPnet and n-LBPnet. Even extracting many handcrafted histograms from faces by varying the LBP neighborhood to characterize them and using a powerful classifier (SVM [11]) for attack detection (all this demanding time), the results of such method are still worse than the ones obtained by LBPnet and n-LBPnet, which work with high-level (deep) features based only on a fixed neighborhood system for LBP calculation. All this indicate that the deep texture features are good source of information for face antispoofing compared to the traditional handcrafted texture features.

Regarding the Accuracy (Acc), Half-Total Error Rate (HTER), False Acceptance Rate (FAR) and False Rejection Rate (FRR), Area Under the Curve (AUC) and Equal Error Rate (EER), Tab. I shows the results of LBPnet, n-LBPnet and other approaches, including some of the already compared and new ones, such as: LPQ (Local Phase Quantization) [20], Gabor wavelets [21], DLTP (Dynamic Local Ternary Pattern) [8], MLPQ/MBSIF (Multiscale LPQ with Multiscale Binarized Statistical Image Features) [9], and CDD (Component Dependent Descriptor) [10]. The higher the Accuracy and AUC and the lower the HTER, FAR, FRR, and EER, the better.

As one can see, the proposed networks, in general, outperformed other techniques, presenting greater results on the NUAA [16] dataset. As for the ROC curves, n-LBPnet outperformed again LBPnet, indicating that the normalization improved the network training. The n-LBPnet architecture obtained the best values in four of six metrics and LBPnet was close to it. MLBP also presented good results due to its multiscale approach. However, it may become impractical.

TABLE I
Results in Different Metrics for LBPnet, n-LBPnet, and Other State-of-the-Art Techniques on the NUAA [16] Dataset. EER and AUC Values Are Usually Obtained From ROC Curves and Acc, HTER, FAR and FRR Rates Are the Ones Obtained When the Threshold of the System Is at 0.5. The Missing Values Were Not Provided by the Authors and the Best Value for Each Metric Is Highlighted

Method	**EER**	**AUC**	**Acc**	**HTER**	**FAR**	**FRR**
LBPnet	0.021	0.993	0.976	0.022	0.028	0.016
n-LBPnet	**0.018**	0.996	**0.982**	**0.017**	0.019	**0.015**
LBP+SVM [7]	0.029	-	-	0.132	-	-
LPQ+SVM [7]	0.046	-	-	-	-	-
Gabor+SVM [7]	0.095	-	-	-	-	-
MLBP [7]	-	0.990	0.980	0.025	**0.006**	0.044
DLTP [8]	-	0.952	0.945	0.035	0.032	0.038
MLPQ/MBSIF [9]	**0.018**	-	-	-	-	-
CDD [10]	0.019	**0.998**	0.977	-	-	-
NUAA Best [16]	-	0.950	0.920	-	-	-
LLD [19]	0.082	0.966	-	-	-	-

Regarding the other accurate methods in Tab. I, the MLPQ/MBSIF and CDD techniques, despite of presenting great results, also combine lots of handcrafted features in order to characterize faces, and may become computationally expensive. The MLBP/MBSIF, e.g., is based on two kind of multiscale handcrafted texture descriptors.

Finally, as can also be observed in Tab. I, the LBP texture descriptor presents itself (with SVM) a good EER, lower than other important descriptors such as LPQ and Gabor, also justifying the choice of integration of such method in LBPnet and n-LBPnet instead of the other techniques.

An advantage of detecting spoofing attacks in simple static images consists in the fact that it usually can be performed in real time and even if there is available only a short image sequence of the face (or only a single image). Besides, there is no need of extra sensors or specific behaviors of the user. All this make the proposed methods even more interesting for real applications.

VI. Conclusion

In this brief, two LBP-based Convolutional Neural Networks, LBPnet and n-LBPnet, are proposed for spoofing detection in face recognition systems, which presented great results on the NUAA spoofing dataset, outperforming other assessed state-of-the-art techniques. With the highest ROC curves, low EER as well as high accuracy, the proposed LBPnet and n-LBPnet networks configure effective alternatives for spoofing detection in real face recognition applications of nowadays. Besides of presenting great results, the proposed methods are more efficient than other state-of-the-art techniques that combine lots of handcrafted information to detect attacks. Our approaches use the LBP descriptor with a single neighborhood, a forward bottom-up pass and simple softmax neurons at the top for detecting spoofing attempts quickly, being more suitable for real time applications. Based on all this it is possible to conclude that deep texture features are rich sources of information for face spoofing detection, propiciating better results than handcrafted ones (or even combination of them, which may become impractical). The integration of the LBP descriptor in a deep learning architecture is a suitable and robust alternative to prevent such criminal activities.

References

[1] A. K. Jain *et al.*, “Biometrics: A grand challenge,” in *Proc. Int. Conf. Pattern Recognit.*, Cambridge, U.K., 2004, pp. 935–942.

[2] M. Fons, F. Fons, and E. Cantó, “Fingerprint image processing acceleration through run-time reconfigurable hardware,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 12, pp. 991–995, Dec. 2010.

[3] D. Menotti *et al.*, “Deep representations for iris, face, and fingerprint spoofing detection,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 864–879, Apr. 2015.

[4] N. K. Ratha, J. H. Connel, and R. M. Bolle, “An analysis of minutiae matching strength,” in *Proc. Int. Conf. Audio Video Based Biometric Person Authentication*, Halmstad, Sweden, 2001, pp. 223–228.

[5] 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.

[6] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[7] J. Määttä, A. Hadid, and M. Pietikäinen, “Face spoofing detection from single images using micro-texture analysis,” in *Proc. Int. Joint Conf. Biometrics*, Washington, DC, USA, 2011, pp. 1–7.

[8] S. Parveen *et al.*, “Face liveness detection using dynamic local ternary pattern (DLTP),” *Computers*, vol. 5, no. 2, p. 10, 2016.

[9] S. R. Arashloo, J. Kittler, and W. Christmas, “Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2396–2407, Nov. 2015.

[10] J. Yang, Z. Lei, S. Liao, and S. Z. Li, “Face liveness detection with component dependent descriptor,” in *Proc. Int. Conf. Biometrics*, 2013, pp. 1–6.

[11] C. Cortes and V. N. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[12] L. Deng and Y. Dong, “Deep learning: Methods and applications,” *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.

[13] Y. Bengio, “Deep learning of representations: Looking forward,” in *Proc. 1st Int. Conf. Stat. Lang. Speech Process. (SLSP)*, Tarragona, Spain, Jul. 2013, pp. 1–37. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-39593-2_1, doi: 10.1007/978-3-642-39593-2_1.

[14] K. Chellapilla, S. Puri, and P. Simard, “High performance convolutional neural networks for document processing,” in *Proc. 10th Int. Workshop Front. Handwrit. Recognit.*, La Baule, France, Oct. 2006. [Online]. Available: https://hal.inria.fr/inria-00112631

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[16] 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. Eur. Conf. Comput. Vis.*, 2010, pp. 504–517.

[17] Y. Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. 22nd ACM Int. Conf. Multimedia (MM)*, Orlando, FL, USA, 2014, pp. 675–678, doi: 10.1145/2647868.2654889.

[18] G. Montavon, G. Orr, and K. R. Müller, *Neural Networks: Tricks of the Trade*, 2nd ed. Berlin, Germany: Springer-Verlag, 2012. [Online]. Available: http://www.springer.com/us/book/9783642352881

[19] W. R. Schwartz, A. Rocha, and H. Pedrini, “Face spoofing detection through partial least squares and low-level descriptors,” in *Proc. Int. Joint Conf. Biometrics*, Washington, DC, USA, 2011, pp. 1–8.

[20] V. Ojansivu and J. Heikkilä, “Blur insensitive texture classification using local phase quantization,” in *Proc. Int. Conf. Image Signal Process.*, Cherbourg-Octeville, France, 2008, pp. 236–243.

[21] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.