JAISCR, 2019, Vol. 9, No. 1, pp. 41 – 49
10.2478/jaiscr-2018-0023

DEEP FEATURES EXTRACTION FOR ROBUST FINGERPRINT SPOOFING ATTACK DETECTION

Gustavo Botelho de Souza[1], Daniel Felipe da Silva Santos[2], Rafael Gonçalves Pires[1], Aparecido Nilceu Marana[2] and João Paulo Papa[2,*]

[1] *UFSCar - Federal University of São Carlos. São Carlos/SP. Brazil. 13565-905*
[2] *UNESP - São Paulo State University. Bauru/SP. Brazil. 17033-360*

E-mail: {gustavo.botelho, danielfssantos1, rafapires}@gmail.com
{nilceu, papa}@fc.unesp.br

Submitted:13th December 2017; Accepted: 20ty December 2017

Abstract

Biometric systems have been widely considered as a synonym of security. However, in recent years, malicious people are violating them by presenting forged traits, such as gelatin fingers, to fool their capture sensors (spoofing attacks). To detect such frauds, methods based on traditional image descriptors have been developed, aiming liveness detection from the input data. However, due to their handcrafted approaches, most of them present low accuracy rates in challenging scenarios. In this work, we propose a novel method for fingerprint spoofing detection using the Deep Boltzmann Machines (DBM) for extraction of high-level features from the images. Such deep features are very discriminative, thus making complicated the task of forgery by attackers. Experiments show that the proposed method outperforms other state-of-the-art techniques, presenting high accuracy regarding attack detection.

Keywords: Restricted Boltzmann Machines, Deep Boltzmann Machines, Deep Learning, Fingerprint Spoofing Detection, Biometrics.

1 Introduction

In the last years, biometric systems became quite common in our activities due to their high security and availability of affordable sensors [1, 2]. However, criminals are already violating them by presenting forged traits, such as gelatin fingers, to fool their capture sensors, a process known as spoofing attack [3]. In this sense, countermeasures techniques must be integrated into the traditional biometric systems to prevent such frauds.

Countermeasure methods proposed so far use, in general, raw, i.e., handcrafted features extracted at the moment of identification, e.g., the presence of facial movement, skin sweat, etc., to detect whether there is a life or a fake biometric trait being presented to the sensor. Such handcrafted methods, however, are shown to be not good enough, especially in challenging scenarios [4].

In this work, we propose a novel approach for spoofing detection in fingerprint recognition systems, an adaptation of the method we presented in

[*] Authors are grateful to grants #2014/12236-1 and #2017/05522-6 from São Paulo Research Foundation (FAPESP), grant #306166/2014-3 from the National Council for Scientific and Technological Development (CNPq), and grant #88881.132647/2016-01 from the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES). Authors are also grateful to Banco do Brasil.

[5], using deep features extracted from images by a probabilistic deep learning architecture: the Deep Boltzmann Machine (DBM) [6]. DBMs can deal with complex patterns efficiently and accurately since they extract and work with high-level features from the original data, being suitable for tasks in which the patterns should not be easily detected or forged. Results on the Crossmatch [4] dataset show that the proposed method outperforms other state-of-the-art techniques, presenting high accuracy regarding attack detection.

2 Technical Background

In this Section, basic concepts regarding spoofing attacks, Restricted Boltzmann Machines (RBM) [7, 8] and Deep Boltzmann Machines (DBM) [6] are presented.

2.1 Biometric Spoofing Detection

In attacks on biometric systems, criminals usually generate synthetic samples of biometric traits of legal users, such as printed facial photographs and gelatin or latex fingers, to fool the capture sensors [9, 10]. Figure 1 shows examples of fingerprints obtained from real and synthetic fingers. As one can observe, even for humans, visually, it is difficult to differentiate between real and fake ones.

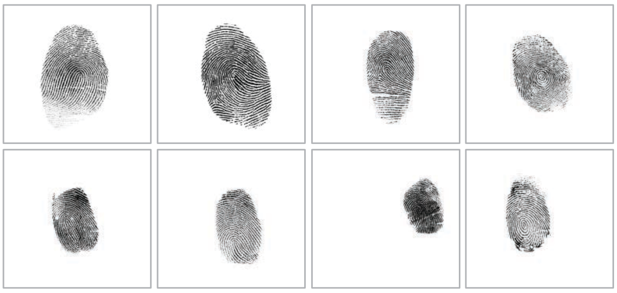


Figure 1. Fingerprints from the Crossmatch 2013 [4] dataset. The top fingerprints are real and the bottom are fake, i.e., obtained from synthetic fingers made of different materials.

Antispoofing methods have been proposed based on different principles. Nevertheless, spoofing detection is still an open question [11]. Most of the techniques are based on simple rules (hand-crafted features) to detect attacks, e.g., the presence of skin sweat. However, criminals can quickly identify these rules and improve attacks: watering the latex fingers. In this sense, algorithms able to work with deep, i.e., high-level and non trivially generated features, are necessary. Among them, the deep learning-based methods simulate the deep structures of neurons in the human brain and have outperformed state-of-the-art techniques in many areas.

2.2 Restricted Boltzmann Machines (RBM)

The Restricted Boltzmann Machines (RBM) [7, 8] are energy-based neural networks used to compose probabilistic deep learning architectures. The model of an RBM (see Figure 2) comprises a visible layer $\mathbf{v}$ with m units and a hidden layer $\mathbf{h}$ with n units. Additionally, a real-valued matrix $\mathbf{W}_{m\times n}$ models the weights between the visible and hidden neurons, where w_{ij} stands for the weight between the visible unit v_i and the hidden unit h_j.

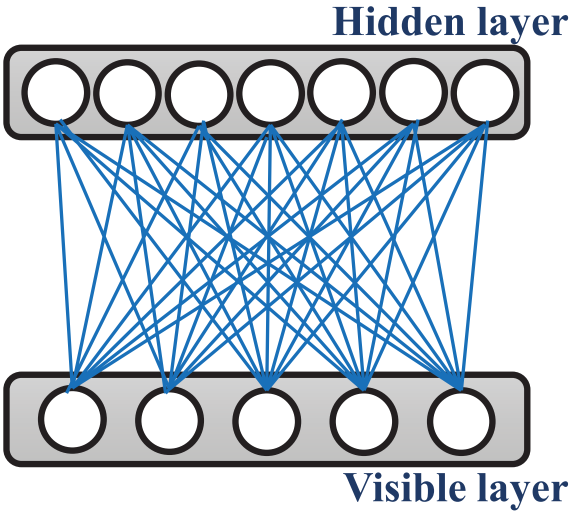


Figure 2. Architecture of a Restricted Boltzmann Machine [8]. Each neuron of a given layer is connected with all the neurons in the opposite layer.

Considering both layers, $\mathbf{v}$ and $\mathbf{h}$, with binary-valued units, i.e., $\mathbf{v} \in \{0,1\}^m$ and $\mathbf{h} \in \{0,1\}^n$, we have the so-called Bernoulli-Bernoulli Restricted Boltzmann Machine (BB-RBM). The energy function of a BB-RBM is given by

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij}, \quad (1)$$

where **a** and **b** stand for the biases of visible and hidden units, respectively.

The marginal probability of a visible configuration (input vector) is given by

$$P(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}, \tag{2}$$

where **Z** corresponds to the so-called partition function.

Since the BB-RBM is a bipartite graph, the activations of both visible and hidden units are mutually independent, thus leading to the following conditional probabilities

$$P(v_i = 1|\mathbf{h}) = \phi\left(\sum_{j=1}^{n} w_{ij}h_j + a_i\right), \tag{3}$$

and

$$P(h_j = 1|\mathbf{v}) = \phi\left(\sum_{i=1}^{m} w_{ij}v_i + b_j\right), \tag{4}$$

where $\phi(\cdot)$ stands for the sigmoid function.

Let $\theta = (\mathbf{W}, \mathbf{a}, \mathbf{b})$ be the set of parameters of a BB-RBM. They are learned through a training algorithm that aims at maximizing the probabilities of occurence of all the available training data (input vectors) $\mathcal{V}$, as follows

$$\arg\max_{\theta} \prod_{\mathbf{v}\in\mathcal{V}} P(\mathbf{v}). \tag{5}$$

One of the most used approaches to solve the above problem is the Contrastive Divergence (CD) [7], which basically ends up performing Gibbs sampling using the training data as the visible units.

In the presence of real-valued data (grayscale images), one should use the so-called Gaussian-Bernoulli RBM (GB-RBM) [12], which now models the input vector as composed of Gaussian units. Therefore, Equation 1 can be reformulated as

$$E(\mathbf{v},\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{m}\frac{(v_i - a_i)^2}{\sigma_i^2} - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{v_i}{\sigma_i}h_j w_{ij}. \tag{6}$$

Since the visible units have been modified, one needs to reformulate their conditional probability. Based on this, Equation 3 can be rewritten as follows

$$P(v_i|\mathbf{h}) = \mathcal{N}\left(v_i\middle|\sum_{j=1}^{n} w_{ij}h_j + a_i, \sigma_i^2\right), \tag{7}$$

where σ^2 stands for the variance of the Gaussian distribution $\mathcal{N}$.

2.3 Deep Boltzmann Machines (DBM)

An RBM [7, 8] can be used for many tasks, such as noise removal. However, to learn more complex and robust representations of the data, a deep architecture is required. A Deep Boltzmann Machine (DBM) [6] consists of a stack of RBMs that learn together. After finding the weights and biases concerning all its layers, such deep neural network can also be used to eliminate noise, pinpointing or to extract deep, i.e., high-level features from data vectors, much more accurately.

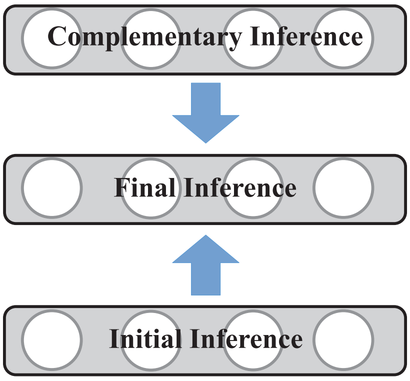


Figure 3. Inference by intermediate field. In DBM, influences from the superior and inferior layers are considered in order to update intermediate layers of the network, forming an undirected network model.

In DBMs, connections among adjacent layers form a complete undirected model: the learning process considers both directions of interaction among adjacent layers, as shown in Figure 3. As one can observe, when analyzing a given layer of the network, its superior layer is considered as the complementary inference, its inferior layer as the initial inference, and the middle layer (layer being analyzed at the moment) as the final inference at the equilibrium stage.

After a bottom-up initialization of the DBM (training one of the stacked RBMs per time, from bottom to top), the learning of the whole DBM is performed through the use of a variable inference method called Mean-Field (MF) to enhance its performance. Such method consists in minimizing the total energy of the whole network according to the parameters found through partial inferences made through the mean-fields (that simplify such process since there are two-way interactions between adjacent layers in the network) [13]. Roughly speaking, the idea is to find an approximation $Q^{MF}(\mathbf{h}|\mathbf{v};\mu)$ that best represents the true distribution of the hidden layers, i.e., $P(\mathbf{h}|\mathbf{v};\theta)$. This approximation is computed through the following factored distribution

$$Q^{MF}(\mathbf{h}|\mathbf{v};\mu) = \prod_{l=1}^{L}\left[\prod_{k=1}^{F_l} q(h_k^l)\right], \tag{8}$$

where L stands for the number of hidden layers, F_l represents the number of nodes in the hidden layer l, and $q(h_k^l = 1) = \mu_k^l$. The goal is to find the parameters of the mean-field $\mu = \{\mu^1, \mu^2, ..., \mu^L\}$.

3 Previous Work

In this Section, we briefly describe an approach that we previously proposed in [5] for fingerprint spoofing detection, also based on a DBM for deep features extraction. Actually, in such previous work and different from the actual one, after learning the parameters of the DBM, a final layer was added at the top of such a structure with two softmax units, forming an MLP (Multilayer Perceptron) network, i.e., a complete classifier, to identify normal (class "0") or attack fingerprint patterns (class "1").

Basically, given an initial training set of grayscale fingerprint images, the first step consisted in the extraction of their relevant regions [5]. After that, resizing and database augmentation techniques were also applied to improve the network performance and avoid lack of data in training. As shown in Figure 4, for each training fingerprint image, its region of interest (ROI) with a fixed size (350 × 231 pixels) was cropped. After that, the ROI was resized to 44 × 29 pixels and 10 different images (patches) with size 36 × 24 were obtained from it. The resultant patches, in amount 10 times greater than the original fingerprint images and with lower dimensions, served as input to train the DBM.

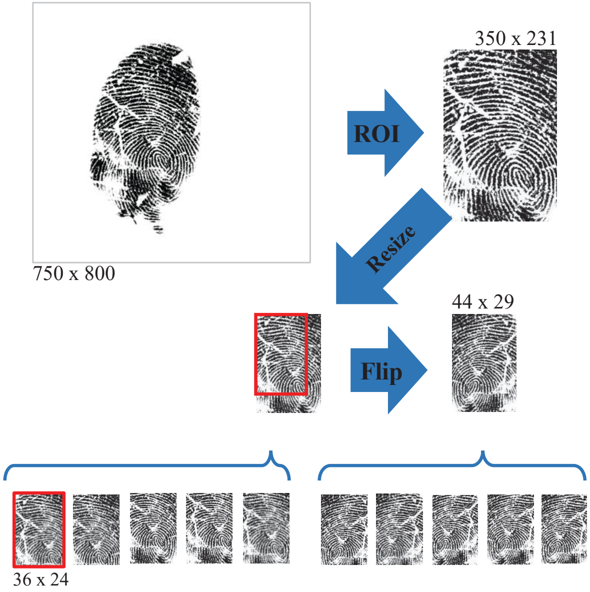


Figure 4. Image normalization and database augmentation process: given each initial high-dimensional training fingerprint image, its ROI was detected, cropped, resized, flipped and 10 different patches were obtained based on the original and flipped ROI, by translating the position of the red rectangle in their four corners and central regions. This process was repeated for all training images and at the end, the patches generated from all of them served as an input to train the DBM [5].

As mentioned in [5], after preprocessing the images from the dataset and before performing the training of the DBM itself, given each grayscale fingerprint patch, it was needed to previously train a Gaussian-Bernoulli RBM (GB-RBM), that served as the interface between such real-valued patches and the Bernoulli-Bernoulli RBMs (BB-RBMs) that constituted the DBM. This previous step with the GB-RBM was called GB Preprocessing and is shown in Figure 5.

After the GB Preprocessing, the DBM bottom-up initialization properly began and the stacked BB-RBMs were trained, in a greedy bottom-up approach, in the same way as the GB-RBM, except by the fact that we used a Bernoulli-Bernoulli sampling approach in the Contrastive Divergence method [7].

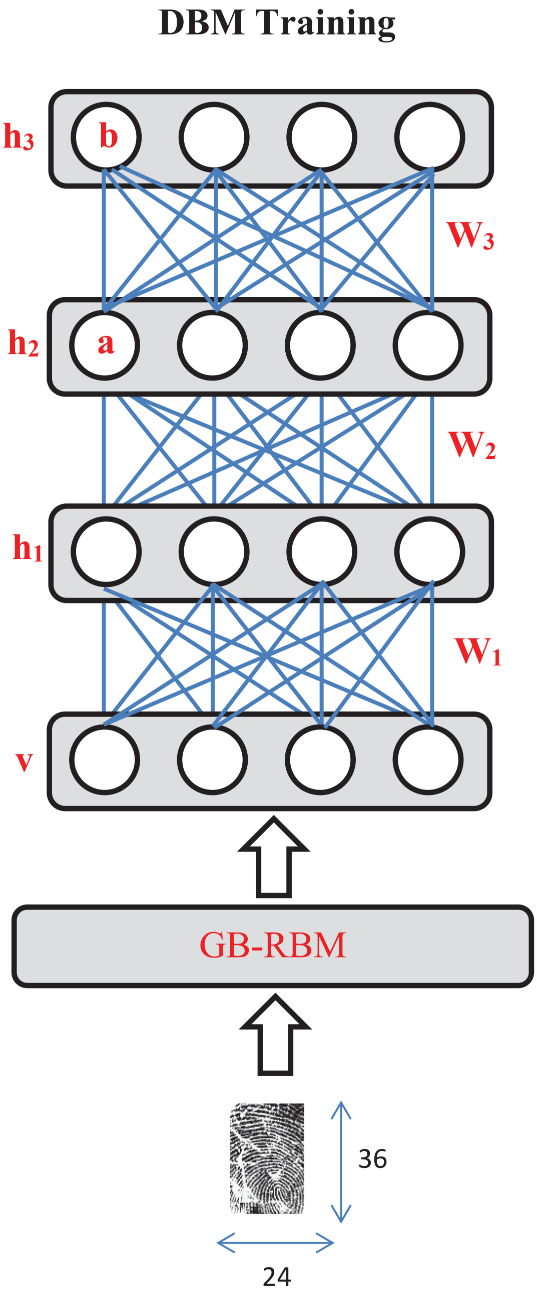


Figure 5. Gaussian-Bernoulli preprocessing followed by a Bernoulli-Bernoulli DBM training based on a fingerprint patch of size 36 × 24 pixels.

Then, after the GB Preprocessing and the bottom-up greedy initialization of the DBM, the Mean-Field algorithm was performed, as in [14]), updating the weights and the biases of the RBMs stacked in the DBM in a more accurate way. Finally, an MLP neural network was constructed (see Figure 6) using the same architecture as the DBM. The initialization of the network weights was conducted using the pretrained parameters, **W** and **b**, of each individually trained RBM. We just added a new top layer with two softmax units for classification (working with a two-class problem, i.e., live or fake images).

The two softmax units were responsible for converting the inputs, coming from the top hidden layer of the DBM into some normalized probabilities, making possible to compare them with the desired network output using the cross-entropy function, that measures the KL divergence [15] between the network and the data probability distributions.

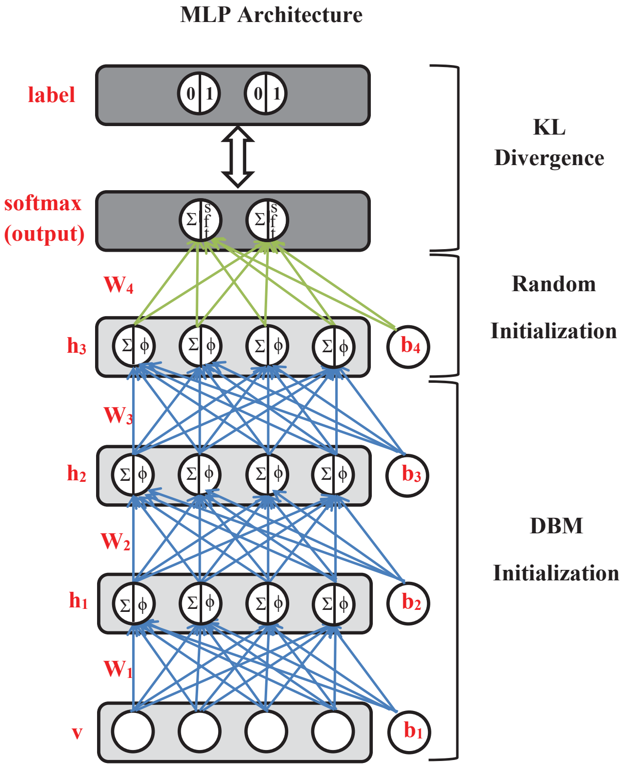


Figure 6. Adjusted parameters **W** and **b** were used to initialize the MLP. This image shows the main functionalities of each layer of the formed network. It is possible to observe that there was a summation and a nonlinear processing has been applied inside each neuron, and it is also possible to observe the softmax units represented in the output layer [5].

Given the formed MLP, the fine-tuning of the network weights was then performed, comparing desired and obtained probabilities by the softmax units, given the training fingerprint patches, and through the conjugate gradient method, making possible to find a better configuration for the parameters **W** and **b** through gradient minimization, as in [16, 17].

In order to classify a test fingerprint image, its 10 patches were extracted, each of them was classified individually by the formed MLP and a votation scheme was performed: after the classification of the 10 patches originated from the test image, the algorithm classified the whole fingerprint as real or fake based on the classification of the majority of its patches (in case of draw, the fingerprint was classified as fake).

4 Proposed Approach

Differently from the previous architecture [5], briefly explained in Section 3, in this work we propose a novel approach for fingerprint spoofing detection based on deep features extracted by a DBM which presents the following training steps: (1) Image Normalization and Database Augmentation; (2) DBM Training; (3) Feature Extraction and SVM (Support Vector Machine) Training. Despite the fact that both methods represent ways of extracting deep features from fingerprint images for spoofing detection, the proposed approach differs from the previous one by not forming an MLP classifier over the DBM structure. Since the training of such classifier can become a complex and expensive task regarding time and processing, in this work, the DBM is used to extract the deep features and, then, feed a traditional SVM [18]. We also consider different levels of abstraction in the DBM structure, i.e., we consider the activation of the neurons in all hidden layers of the DBM in the composition of the feature vectors of the fingerprint images.

4.1 Image Normalization and Database Augmentation

In this first step, as in the previous approach and as shown in Figure 4, for each training fingerprint image from the Crossmatch [4] dataset, its region of interest (ROI) with a fixed size (350×231 pixels) is cropped by finding the center of mass of the fingerprint pixels. After that, the ROI is resized to 44×29 pixels and 10 different images (patches) with size 36×24 are obtained from it. Such a process is performed for all the training fingerprint images, and the resultant patches will serve as input to train the DBM.

In order to find the original ROI, the image is binarized and a closing operation with a squared structure of size 21×21, adequate to the database, is applied in order to eliminate eventual noise from the sensor and to make the fingerprint a single connected region. After that, to find the position of the fingerprint in the image (that may vary since the users may position their fingers in different parts of the sensor area), the center of mass of the resulting binary image is calculated. Then, based on the center of mass, the 350×231-sized window is cropped from the original image, and its ten final patches are generated and used in the DBM training step.

4.2 DBM Training

Before performing the training of the DBM itself, which will be used for the deep features extraction, given each grayscale fingerprint patch, a Gaussian-Bernoulli RBM (GB-RBM) [12] is also trained in this new approach in order to convert the real-valued patches to posterior probabilities of activation, which will feed the Bernoulli-Bernoulli RBMs (BB-RBM) of the DBM. This previous step with the GB-RBM, called GB Preprocessing, can be observed in Figure 5. The input data (fingerprint patches) are normalized to have zero-mean and unitary variance, allowing to get rid of the σ^2 term from Equations 6 and 7, is not necessary to learn such parameter during training and simplifying such process.

After the GB Preprocessing, the DBM bottom-up initialization properly begins: each training patch is presented to the Gaussian-Bernoulli RBM, and its posterior probabilities values feed the visible layer of the first stacked BB-RBM. After training each BB-RBM of the DBM stack, one per time from bottom to top, the Mean-Field algorithm is also applied, updating the weights and the biases of the DBM in a whole and accurate way.

4.3 Feature Extraction and SVM Training

After learning the parameters of the GB-RBM and the DBM, each training fingerprint patch is presented again to such structures, and a forward pass through all these layers is performed. The feature vector of the given patch is composed by concatenating all activation probabilities of the neurons in all the hidden layers. These values incorporate high-level statistical features from the original image.

After finding the feature vectors of all training samples, a dimensionality reduction is performed through PCA [19] (Principal Component Analysis) preserving almost 98% of original information while reducing their length to 140 positions. Given all reduced training feature vectors, an SVM classifier with radial basis function kernel is trained to identify attempts of spoofing. As an observation, in the grid search of SVM, only 25% of the training feature vectors (randomly selected), in a 10-fold cross-validation scheme, were considered for efficiency. Figure 7 shows the proposed architecture.

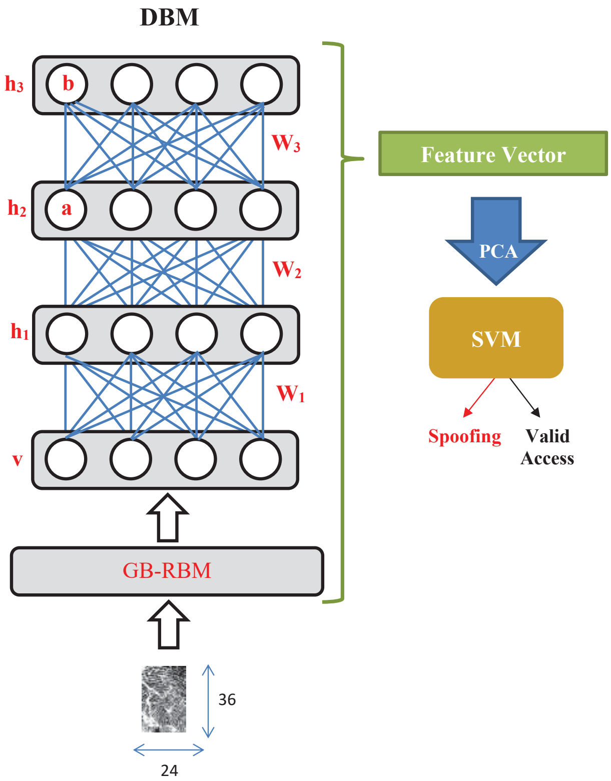


Figure 7. Proposed architecture for fingerprint spoofing detection. After Gaussian-Bernoulli Preprocessing, Bernoulli-Bernoulli DBM and SVM training, test fingerprint patches can be classified based on the activation of the neurons of the trained network.

After finding the weights for the DBM and training the SVM, such models can be used for the spoofing detection task. Given an unknown fingerprint image, its ten patches are extracted and classified as in training phase. Given the classification of its ten patches, the majority of votes determines the final class of the test fingerprint: from the real or synthetic finger (in case of a draw, the fingerprint is taken as synthetic).

5 Experiments, Results, and Discussion

In our experiments, we used an architecture and training parameters similar to [5] to allow a fair comparison of the methods. We used a GB-RBM with 864 (patch of size 36 × 24) visible units and 1,000 hidden ones. Over it, we stacked a DBM composed by two BB-RBMs with 1,000 visible and 1,000 hidden units each. The GB-RBM was trained over 500 epochs and the BB-RBMs over 200 epochs each, from bottom to top. After that, 200 epochs were applied in the Mean-Field algorithm (30 iterations per epoch). For the GB-RBM, we used a learning rate of 0.001, a momentum of 0.5 (and 0.9 after the fifth epoch), and weight-decay of 0.0002. In the case of BB-RBMs, we used a learning rate of 0.01, the same momentum and weight-decay values. Sucha a lower learning rate for the GB-RBM is required by the fact that the activation function of the neurons of its visible layer is not bounded as in binary neurons. The Mean-Field learning rate was also 0.001.

The training and testing algorithms were implemented in C and Matlab, using CUDA and the libraries BLAS and OpenCV. To accomplish this work, we used an Intel-i7 laptop with eight cores and 8 GB of RAM. The graphic hardware used was a Geforce GT650M board having 385 cores, 2 GB of dedicated memory and two streaming processors. We also applied LIBSVM to train the SVM classifier, which supports multiprocessing, making the training even faster.

We evaluated the proposed approach on the Crossmatch [4] database from the LivDet 2013 competition, considered by the own authors the most challenging dataset of the event. There are 2,250 images of fingerprints for training and 2,250 images for testing, 1,250 from real fingers and 1,000 from fake ones, in each set. Figure 1, in Sec. 2.1, shows examples of fingerprints from the Crossmatch database.

The results regarding Accuracy (ACC), False Acceptance Rate (FAR) and False Rejection Rate (FRR) of the proposed approach and the other state-of-the-art methods are shown in Table 1. The proposed method outperformed all other techniques concerning FRR, including our previous work, recently proposed in [5], which also uses a DBM as its main learning structure. Such a method, briefly explained in Section 3, presents a much more complex algorithm due to the fine tuning step of the formed MLP, which requires backpropagation. In the proposed approach, no fine-tuning step on the network is performed.

The proposed method also presented a close ACC value to the one obtained by such previous method, and much better than all other evaluated techniques, as well as it also presented a low FAR. Additionally, it is important to note that the methods that presented FAR as of 0.00% also presented almost 100.00% of FRR, i.e., they rejected all the fingerprints, which means they may be not suitable for real situations.

Table 1. Results (%) in different metrics of the proposed method, our previous work [5] and other state-of-the-art techniques (obtained from [4]). The FAR and FRR rates are the ones when the threshold of the system is at 0.5. The best value in each metric is highlighted.

Method	FAR	FRR	ACC
Proposed Approach	20.70	**8.96**	85.82
Previous Work	19.40	9.76	**85.96**
Dermalog	**0.00**	99.84	44.53
Anonym1	2.40	86.96	50.53
ATVS	10.30	90.40	45.20
Anonym2	0.30	98.40	45.20
UniNap1	31.10	31.28	68.80
UniNap2	48.30	55.20	47.87
UniNap3	48.30	55.20	47.87
Anonym3	0.10	95.52	46.89
HZ-JLW	**0.00**	100.00	44.44
Itautec	13.90	64.96	57.73
CAoS	54.20	41.92	52.62

6 Conclusion

In this work, we presented a novel approach for fingerprint spoofing detection based on the Deep Boltzmann Machine, which deals with complex patterns in an accurate way due to its probabilistic multilayer architecture. In the proposed method, after training a DBM, such structure can be used to extract deep (high-level) features of the images. An SVM classifier is fed with the feature vectors of the images generated by the DBM to identify spoofing attacks. The proposed approach, due to its deep architecture, is very robust, outperforming the state-of-the-art techniques assessed on the Crossmatch dataset. Besides, DBMs can be trained in a non-supervised way to extract high-level features of an input data, a very suitable model for real applications (only the classifier, e.g., an SVM, needs some labeled samples).

References

[1] A. Jain, A. Ross and K. Nandakumar, Introduction to Biometrics, Springer, 2011.

[2] A Biniaz and A. Abbasi, Segmentation and edge detection based on modified ant colony optimization for iris image processing, Journal of Artificial Intelligence and Soft Computing Research (JAISCR), vol . 3, no. 2, 2013, pp. 133-141.

[3] D. Menotti, G. Chiachia, A. Pinto, W. Schwartz, H. Pedrini, A. Falcao and A. Rocha, Deep representations for iris, face, and fingerprint spoofing attack detection, IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, 2015, pp. 864-879.

[4] L. Ghiani, V. Mura, S. Tocco, G. Marcialis, F. Roli, D. Yambay and S. Schuckers, LivDet 2013 fingerprint liveness detection competition, In: Proceedings of International Conference on Biometrics, 2013, pp. 1-6.

[5] G. Souza, D. Santos, R. Pires, A. Marana, J. Papa, Deep Boltzmann Machines for robust fingerprint spoofing attack detection, In: Proceedings of International Joint Conference on Neural Networks, 2017, pp. 1863-1870.

[6] R. Salakhutdinov and G. Hinton, Deep Boltzmann Machines, Technical Report, University of Toronto, 2009.

[7] G. Hinton, Training products of experts by minimizing Contrastive Divergence, Neural Computation, vol. 14, no. 2, 2002, pp.1771-1800.

[8] G. Hinton, Neural networks: tricks of the trade, Springer, Berlin, 2012.

[9] N. Ratha, J. Connel and R. Bolle, An analysis of minutiae matching strength, In: Proceedings of International Conference on Audio- and Video-Based Biometric Person Authentication, 2001, pp. 223-228.

[10] J. Galbally, J. Fierrez and J. Garcia, Vulnerabilities in biometric systems: attacks and recent advances in liveness detection, Database, vol. 1, no. 3, 2007, pp. 1-8.

[11] K. Patel, H. Han and A. Jain, Cross-database face antispoofing with robust feature representation, In: Proceedings of Chinese Conference on Biometric Recognition, 2016, pp. 611-619.

[12] V. Nair and G. Hinton, Implicit mixtures of Restricted Boltzmann Machines, Advances in Neural Information Processing Systems, vol. 21, 2009, pp. 1145-1152.

[13] D. MacKays, Information theory, inference and learning algorithms, Cambridge University Press, 2003.

[14] R. Salakhutdinov and H. Larochelle, Efficient learning of Deep Boltzmann Machines, Artificial Intelligence and Statistics, 2010, pp. 693-700.

[15] S. Kullback, Probability densities with given marginals, Annals of Mathematical Statistics, vol. 39, no. 4, 1968, pp. 1236-1243.

[16] I. Navon and D. Legler, Conjugate-gradient methods for large-scale minimization in Meteorology, Monthly Weather Review, American Meteorological Society, vol. 115, 1987, pp. 1479-1502.

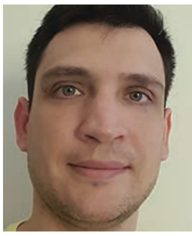
[17] Y. LeCun, L. Bottou, G. Orr and K. Müller, Efficient Backprop., Springer-Verlag, United Kingdom, 1998.

[18] C. Cortes and V. Vapnik, Support-vector networks, Machine Learning, vol. 20, no. 3, 1995, pp. 273-297.

[19] H. Hotelling, Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology, vol. 24, 1933, pp. 417-441.

Gustavo Botelho de Souza is an IT Technician (2007), B.Sc. (2010) and M.Sc. (2013) in Computer Science by the São Paulo State University (UNESP). Ph.D. candidate in Computer Science at Federal University of São Carlos (UFSCar) since 2015 with visiting period at Michigan State University (Biometrics Research Group) in 2017. Researcher at Banco do Brasil, his research interests include Image Analysis, Biometrics and Pattern Recognition. His Ph.D. thesis concerns in the application of deep learning architectures to biometric spoofing detection.

Daniel Felipe da Silva Santos is a B.Sc. (2014) and M.Sc. (2017) in Computer Science by the São Paulo State University. He developed an automated vehicle classifier based on outdoor images for monitoring public roads and his research interests involve Image Analysis, Visual Surveillance and Deep Learning.



Rafael Gonçalves Pires is a B.Sc. (2009) in Computer Science by Universidade do Sagrado Coração (USC), M.Sc (2014) in Computer Science by São Paulo State University, and Ph.D candidate in Computer Science at Federal University of São Carlos since 2014. His research interests includes Pattern Recognition, Image Denoising and Restoration.

Aparecido Nilceu Marana is a B.Sc. (1986) in Mathematics by the São Paulo State University and M.Sc. (1990) in Computer Science by the State University of Campinas (UNICAMP). He got his Ph.D. degree in Electrical Engineering from UNICAMP and King's College (London), in 1997, and has experience in Image Analysis, Biometrics and Computer Vision. He is an Associated Professor at the Department of Computing at UNESP, campus of Bauru. In 2005, he was a visiting researcher (postdoc fellow) at PRIP Laboratory (Michigan State University). Coordinator of the Computer Science Graduate Program (UNESP) from 2010 to 2014.

João Paulo Papa is a B.Sc. (2002) in Information Systems by the São Paulo State University, M.Sc. (2005) in Computer Science by the Federal University of São Carlos, and Ph.D. in Computer Science (2008) from the State University of Campinas. He was a post doctoral fellow in Computer Science by the State University of Campinas and Harvard University (2015). Currently, he is an Associate Professor at the Department of Computing at São Paulo State University, campus of Bauru. His main research interests include Image Analysis, Pattern Recognition and Medical Diagnosis assisted by computer systems. He is the actual Coordinator of the Computer Science Graduate Program (UNESP) and member of the editorial board of important journals such as the IEEE Signal Processing Letters and Computers and Electrical Engineering.