Efficient Deep Learning Architectures for Face Presentation Attack Detection

Gustavo Botelho de Souza[1,2,*], João Paulo Papa[1,3] and Aparecido Nilceu Marana[1,3]
[1]*Federal University of São Carlos (UFSCar) - São Carlos/SP - Brazil*
[2]*IT Board - Banco do Brasil - Brasília/DF - Brazil*
[3]*São Paulo State University (UNESP) - Bauru/SP - Brazil*
E-mail: gustavo.botelho@gmail.com, {joao.papa, nilceu.marana}@unesp.br

***Abstract*—Biometric systems are common in our everyday life: from our mobile devices to huge surveillance systems. Despite the higher difficulty to circumvent biometric applications, criminals are simulating traits such as face or fingerprints of valid users (presentation attacks - PA), in order to fool the security applications. Deep neural networks have obtained state-of-the-art results in PA detection. However, in many cases, they are computationally expensive, being not feasible in environments with hardware restrictions, such as mobile ones. In this sense, we propose efficient deep learning architectures for PA detection, especially for face recognition systems, able to be trained and deployed even when there is low computational power available.**

I. Introduction

Biometrics emerged as a robust solution for security systems, recognizing people by "who they are". [Jain et al. 2011]. Despite the higher difficulty in circumventing the biometric applications when compared with traditional security systems based on passwords or token, criminals are already developing techniques to simulate biometric traits of legal users, such as using printed photographs to simulate valid faces (process known as spoofing or presentation attack - PA), to fool the security mechanisms, especially in commercial applications [ISO 2016].

Face is a promising biometric trait for our days given its universality, non-intrusive and fast capture, by means of common digital cameras, which are available, nowadays, almost everywhere. The traditional face recognition systems are the ones that most suffer with PAs given the high availability of photographs of people in the world wide network, especially in social networks. All this makes PA detection techniques essential to security systems based on faces.

Convolutional Neural Networks (CNN) [LeCun et al. 1998] have presented the best results in many tasks, including biometric recognition and PA detection such as in [Atoum et al. 2017], obtaining good accuracy rates. However, most of the deep learning models are extremely computationally expensive, requiring specialized hardware for training and deployment (powerful GPUs) as well as lots of data and storage capacity, which are are not available in mobile environments or even in developing countries.

Main contributions: (i) proposal of alternative deep neural networks (not only CNNs) and demonstration of their effectiveness without the usage of GPUs; **(ii)** proposal of novel texture based CNN architectures, demonstrating the superiority of deep texture features when compared with handcrafted ones; **(iii)** proposal of Partial Learning, a novel training algorithm for CNNs based on deep local features for face PA detection; **(iv)** development of proprietary deep learning based systems for face recognition and PA detection to Banco do Brasil; **(v)** presentation and publication of academic and technical papers in qualified conferences and journals; **(vi)** awards, such as the IEEE Computational Intelligence Society Student Travel Grant Award and the 2^{nd} place in the the Face Recognition Challenge at the International Summer School for Advanced Studies on Biometrics; and **(vii)** dissemination of knowledge and results obtained in important events, such as Brazilian Conference on Intelligent Systems (BRACIS 2018) - paper presentation and invited tutorial on Deep Learning.

II. Proposed RBM Based Model

The first approaches proposed in this work were based on highly efficient neural network models called Restricted Boltzmann Machines (RBM) [Hinton 2002], which can be trained using low computational power, usually single CPUs. Restricted Boltzmann Machines [Tang et al. 2012], [Hinton 2012] are energy-based stochastic neural networks composed of two layers of neurons (visible and hidden), in which the learning phase is conducted by means of an unsupervised fashion. The RBM, actually, is based on the classical Boltzmann Machines [Ackley et al. 1985] with the restriction that no connections between neurons of the same layer are allowed. This restriction allows training in a significantly lower complex way with no high loss in terms of accuracy.

Let $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b})$ be the set of parameters (weights and biases, respectively) of an RBM, they can be learned though a training algorithm that aims at maximizing the product of ocurrence probabilities given all the available training data $\mathcal{V}$, as follows:

*PhD Thesis in Computer Science - This work was sponsored by Banco do Brasil, CAPES (grant PDSE #88881.132647/2016-01), FAPESP (grants #2014/12236-1, #2016/19403-6, #2017/05522-6), NVidia and Michigan State University (doctoral visiting period at the "Biometrics" and "PA Detection" research groups).

$$\arg\max_{\theta} \prod_{\mathbf{v}\in\mathcal{V}} P(\mathbf{v}). \quad (1)$$

One of the most used approaches to solve the above problem is the Contrastive Divergence (CD) [Hinton 2002], which basically ends up performing Gibbs sampling using the training data as the visible units.

We can modify a generative RBM in order to turn it into a classifier. To do so, one of the possibilities is to include especial neurons in the visible layer in order to identify the class of the presented input signal (calling them DRBM - Discriminative RBM). Given each input training data, its class is informed as the activation of the respective special neuron (the other special neurons stay in zero) and the network is trained to correctly activate such special neuron. After presenting an unknown pattern to the RBM and performing its forward and backward pass, the activations of the special neurons indicate the class of the test pattern. This can be used in order to identify real and fake face.

We also can stack many RBMs and perform classification at the top one, as we propose and demonstrate in Fig. 1. In this case, we train each RBM from bottom to top without considering labels, and then train the top one with the labels of the faces. Tab. I shows the results obtained using a single DRBM and the proposed DDRBM (Deep DRBM) in classificating real and fake faces demonstrating that the deep features of the stacked approach allow better results - the DDRBM obtained better accuracy results than the single DRBM and than the state-of-the-art result so far for the the NUAA [Tan et al. 2010] dataset as well as a much lower standard deviation (more stable performance of the stacked approach).

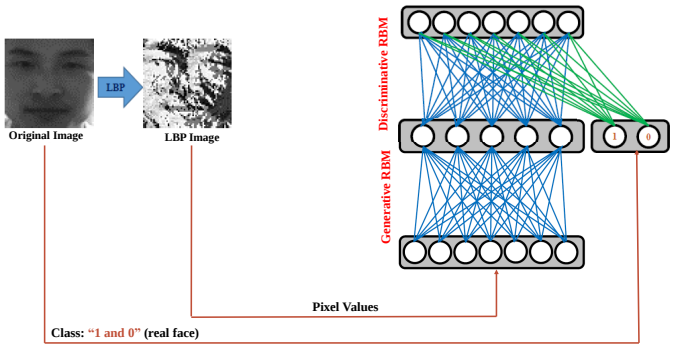

Fig. 1. Proposed DDRBM: generative RBM at the bottom and discriminative RBM at the top for classifying faces in real or fake.

TABLE I
RESULTS FOR THE DRBM AND DDRBM ON THE NUAA DATASET.

Method	Accuracy	Std. Dev.
NUAA Best	0.920	-
DRBM	0.913	0.012
DDRBM	**0.923**	**0.008**

III. PROPOSED TEXTURE BASED CNN ARCHITECTURES: LBPNET AND N-LBPNET

We also proposed a novel CNN architecture, called LBPnet, by integrating the LBP (Local Binary Patterns) [Ojala et al. 2002] descriptor, a robust texture descriptor for face PA detection, in the first layer of a convolutional neural network, in order to extract deep texture features, instead of handcrafted texture histograms, from the facial images for a more robust PA 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.

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 (real or fake face) using the softmax function. Given a detected and normalized grayscale facial image (in this work 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.

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).

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.At the top, 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.

An extended version of LBPnet, called normalized LBPnet (n-LBPnet), was also proposed. The n-LBPnet architecture is quite similar to the LBPnet model, however a Local Response

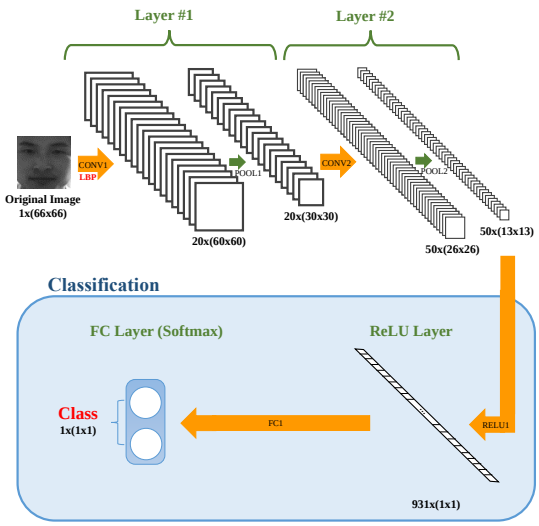

Fig. 2. Architecture of LBPnet. The first two layers perform convolution and pooling operations, (*CONV1*, *POOL1*) and (*CONV2*, *POOL2*), respectively. Given an original image with size 66×66, the number of output feature maps and their sizes, after each operation, are shown below their representations.

Normalization (LRN) [Krizhevsky et al. 2012] step, which simulates the competitive process presented by neurons of human brain in nearby areas, is present between the convolution and pooling operations in the second layer of the CNN.

A. Experiments, Results and Discussion

The proposed networks, LBPnet and n-LBPnet, were assessed on the traditional NUAA Photograph Imposter Database [Tan et al. 2010], 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. The normalized images (in grayscale and with size of 64×64) were already provided by the authors of the database 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. Regarding the ROC curves, Fig. 3 shows their True Acceptance Rate (TAR) *versus* the False Acceptance Rate (FAR) compared with other state-of-the-art-methods: (i) n-LBPnet; (ii) LBPnet; (iii) the MLBP-based method [Chingovska et al. 2012]; (iv) the best method of the original paper of the NUAA [Tan et al. 2010] 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) [Schwartz et al. 2011] 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 [Tan et al. 2010] database and the LLD approach [Schwartz et al. 2011] by far (both based on handcrafted texture features), presenting considerably higher curves.

Despite the MLBP-based approach [Chingovska et al. 2012] 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 - Support Vector Machine [Cortes and Vapnik 1995]) 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.

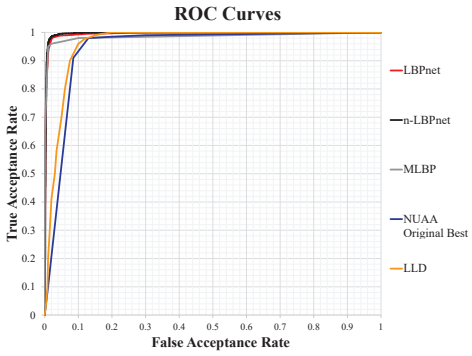

Fig. 3. ROC curves of the proposed CNNs and other state-of-the-art approaches. The higher the curve, the better the method.

IV. Partial Learning - A Novel Training Algorithm for CNNs Applied to PA Detection

We proposed a novel training algorithm, called Partial Learning, for face PA detection, applied to traditional CNN architectures, therefore called lsCNN (Locally Specialized CNN), for a more effective learning of deep local attack features, based on two steps: (i) the Local Pre-training phase, in which each part of the model is trained on each main facial region (predefined and fixed), learning deep local features for attack detection from such areas, and allowing to initialize the whole model in a great better position in the search space; and (ii) the Global Fine-tuning phase, in which the whole model

is fine-tuned based on the weights learned independently by its parts on the facial regions, in order to improve its generalization.

A. lsCNN Architecture

Basically, the proposed lsCNN presents 4 convolutional and pooling layers ($Conv1/Pool1$ to $Conv4/Pool4$) at the bottom. They are followed by a fully-connected layer ($FC1$), a batch normalization and ReLU layers, as well as a dropout one ($Drop1$). At the top, there is a softmax layer with two neurons to classify the faces in real or fake. Each convolutional layer is immediately followed by a batch normalization and signal rectification (ReLU - Rectified Linear Unit) layer. The batch normalization layer serves to normalize the output feature maps obtained in the convolutional layers, improving learning. The rectification function, in each neuron, acts as its activation function, eliminating negative values in the resultant feature maps and also accelerating training.

1) Local Pre-training: In order to initialize the whole lsCNN model in a better position in the search space and make it specialized in deep local attack features from each region of the faces, we split each training face into 9 main regions (patches), regions also adopted for face recognition. After this, we also split the lsCNN architecture into 9 independent smaller CNNs, called PatchNets for simplicity, presenting, each of them, a ninth of the size of the complete original model, and being trained on each of the 9 main facial regions considered from the faces, from $p1$ to $p9$. Each PatchNet had as input RGB patches with 32×32 pixels from a respective region of the training faces. Fig. 4 illustrates the training process of the 9 instances of this smaller neural network on the facial regions of a given image. As one can observe, on the top of each PatchNet there are two softmax neurons since they are trained to classify their respective patches as being real or fake.

2) Global Fine-tuning: After training the 9 smaller neural networks on their respective facial regions, their weights and biases are used to initialize the parts of the whole lsCNN for a fine-tuning step of such larger model on the whole training facial images, in order to improve its generalization. As shown in Fig. 5, each smaller network initializes the weights of the connections and biases of a partition (a ninth) of the lsCNN model, from the left (top) to the right (bottom) side of the lsCNN model. The weights of the first PatchNet, e.g., initialize the connections between the most left neurons of the lsCNN model, responsible for first feature maps (from $FM1-FM3$, in the first layer, to $FM1-FM4$, in the second layer), and so on. The connections of lsCNN between neurons from different parts of it are zero-initialized.

The weights of the two fully-connected layers on top are randomly initialized from a normal distribution in order to improve the generalization of model even more. Their biases are zero-initialized. In Fig. 5, for simplicity, in each partition of lsCNN, only the connections from a neuron in a given feature map to the neurons of the previous layer are shown, as well as the connections of the selected neurons in the first part of lsCNN to their receptive fields in the other parts of such

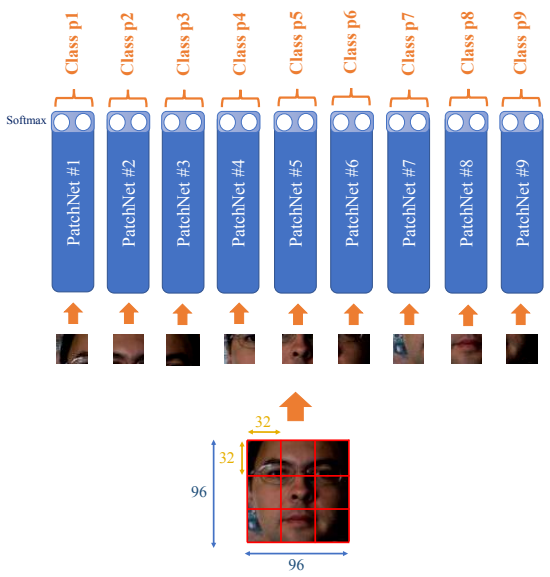

Fig. 4. Illustration of the local pre-training process of lsCNN. Given a facial image, it is split into its 9 main regions, from $p1$ to $p9$, and 9 instances of the smaller CNN architecture (PatchNet) are trained on each of them.

whole model. However, the lsCNN has all the connections of a traditional CNN. 2 After the initialization, the same training facial images (which were split into patches in the former step) are used to fine-tune the weights of the whole lsCNN model, also allowing it to detect some global or more generic features from whole faces, which were not learned locally in the pre-training step.

B. Experiments, Results and Discussion

We evaluated the proposed lsCNN architecture on larger databases: (i) Replay-Attack [Chingovska et al. 2012] dataset; and (ii) CASIA FASD (Face Antispoofing Database) [Zhang et al. 2012].

Regarding Replay-Attack [Chingovska et al. 2012] and CASIA [Zhang et al. 2012] datasets, the lsCNN obtained much better results in terms of ERR and HTER than the traditionally trained CNN and state-of-the-art methods. In order to allow a more robust analysis of lsCNN, we performed larger experiments on the Replay-Attack [Chingovska et al. 2012] and CASIA [Zhang et al. 2012] databases. The Replay-Attack dataset contains 360 videos for training, 360 videos for validation and a test set with 480 videos. The CASIA [Zhang et al. 2012] dataset presents videos of 50 subjects, 12 videos per subject being 3 of real faces and 9 of fake faces. The dataset is divided in training set (20 subjects, 240 videos) and test set (30 subjects, 360 videos). There is no validation set explicitly defined for this database.

In the experiments on both datasets, in order to classify a video, we considered a majority of votes scheme of the faces in its frames. Frames with no face detected by the MTCNN architecture were discarded.

Unlike the experiment with the NUAA dataset, in the experiments with the Replay-Attack and CASIA databases,

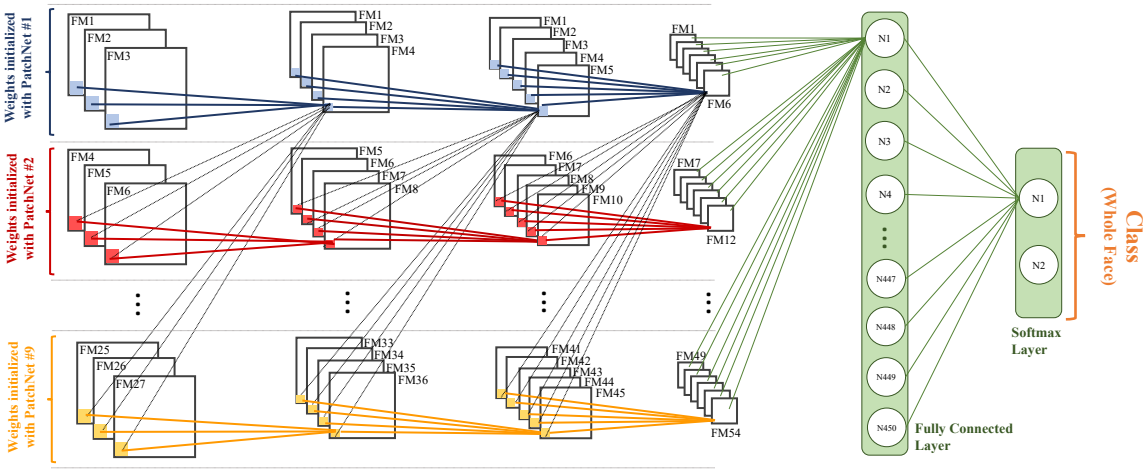

Fig. 5. Initialization of lsCNN based on the weights of the 9 PatchNets. The thicker colored lines represent 3×3 connections and are initialized with the weights learned by each PatchNet. The first PatchNet, e.g., initializes the weights between the first neurons (first feature maps - FM) in the layers of lsCNN. The thin black dotted lines also indicate 3×3 connections zero-initialized and the green thin ones are initialized with random values from a normal distribution (zero-mean and standard deviation of 0.01, by default). The thin gray lines are just for a better visualization of the initialization process.

we considered the original architecture of lsCNN given the larger facial images obtained. After cropping the faces of all frames of all training videos, an augmentation process on both datasets was performed. In each of them, initially and for each facial image, we generated two new versions of it by increasing or decreasing the values of the R, G, and B channels by 50. This was done in order to force the neural network to not rely on brightness for spoofing detection (we did not apply techniques for attenuating the shadows on the faces since they are important to distinguish real faces from 2D fake faces).

For each of the three versions of each original training facial image, we also applied noise or blur transformations in three levels each (with low magnitudes to not affect the images too much), in order to make the neural network also learn smoother features and not rely much on noise. Again we used the Matlab toolbox for applying blur and Gaussian noise to the images. The blur operation was applied in three levels (using a 2×2 Gaussian filter with standard deviations of 0.1, 0.5 and 1.0), as well as the Gaussian noise (with standard deviations of 0.00005, 0.000075 and 0.0001). Such transformations were applied isolatedly, so we obtained, for each of the three initial images from a given face, 6 representations of it. In this sense we augmented our dataset 19 times (original images and $3 \times 6 = 18$ transformed images).

For the Replay-Attack dataset we obtained $1,766,031$ training facial images, and for the CASIA dataset, $852,568$ images. Again, we initialized all weights of the smaller PacthNets based on random values from a zero-mean normal distribution (standard deviation of 0.0001) and normalized each channel of the input facial images by subtracting the mean value of it and diving all the image values by 128 (before splitting them), in order to ensure that most of them would belong to the interval $[-1; 1]$. The biases of the neurons were all zero-initialized. As optimizer, we also used the Adam [Kingma and Ba 2015] method in both cases, with the same following parameters: 64 training images per batch, base learning rate of 0.0001, first momentum of 0.9 and second momentum of 0.999.

In both experiments, we trained the 9 smaller PatchNets for $5,000$ iterations on the facial patches using the Caffe [Jia et al. 2014] framework and initialized the whole lsCNN model. Then we fine-tuned it over $100,000$ iterations. For the Replay-Attack dataset, the best model was obtained (considering results on the validation set of videos) on iteration $53,600$. For the CNN with the same architecture, traditionally initialized with random values extracted from a normal distribution with zero-mean and standard deviation of 0.0001 (biases also zero-initialized) and trained on the whole faces, the best model was obtained only on iteration $74,200$ (much later). The results of the proposed approach and of state-of-the-art methods are presented in Tab. II. For simplicity, we denoted the traditionally trained CNN with the same architecture of lsCNN as "lsCNN Traditionally Trained".

TABLE II
RESULTS ON REPLAY-ATTACK [CHINGOVSKA ET AL. 2012] DATASET: EQUAL ERROR RATE (EER) ON THE VALIDATION DATASET AND HALF-TOTAL ERROR RATE (HTER) ON THE TEST SET. BEST VALUES ARE HIGHLIGHTED.

Method	EER	HTER
Efficient Fine-Tuned VGG-Face [Souza et al. 2017]	—	16.62
Patch Based Handcrafted Approach [Akhtar and Foresti 2016]	—	5.0
Whole Fine-Tuned VGG-Face [Lucena et al. 2017]	—	**1.20**
Fine-Tuned VGG Face [Li et al. 2016]	8.40	4.30
Li et al. [Li et al. 2016]	2.90	6.10
Random Patches Based CNN [Atoum et al. 2017]	2.50	**1.25**
MobileNet-v1 [Howard et al. 2017]	1,67	3,13
Boulkenafet et al. [Boulkenafet et al. 2015]	0.40	2.90
lsCNN Traditionally Trained	**0.33**	1.75
lsCNN	**0.33**	2.50

As one can observe, besides obtaining the best EER, lsCNN presented a great HTER, much lower than expensive methods,

which work with extremely complex and large CNNs, such as VGG-Face [Parkhi et al. 2015]. Despite obtaining a worse HTER result than the traditionally trained neural network, lsCNN obtained the presented results much faster (in a much earlier iteration of the training), as mentioned.

Regarding the CASIA experiment, the best model for lsCNN was obtained on iteration $9,800$, while the best model for the traditionally trained CNN was obtained on iteration $80,900$. In order to compare the performances of such methods with state-of-the-art approaches, we measured the EER, since this dataset presents a predefined test dataset. Tab. III shows the results.

TABLE III
RESULTS IN THE CASIA [ZHANG ET AL. 2012] DATASET OF THE PROPOSED NETWORK ARCHITECTURE (LSCNN) AND OTHER STATE-OF-THE-ART METHODS. THE BEST VALUES ARE HIGHLIGHTED.

Method	EER
Fine-tuned VGG-Face [Li et al. 2016]	5.20
LSTM-CNN [Wang et al. 2018]	5.17
Yang et al. [Yang et al. 2014]	4.92
Patch Based Handcrafted Approach [Akhtar and Foresti 2016]	4.65
Li et al. [Li et al. 2016]	4.50
Random Patches Based CNN [Atoum et al. 2017]	**4.44**
lsCNN Traditionally Trained	**4.44**
lsCNN	**4.44**

As one can observe, lsCNN obtained the best EER on the CASIA dataset, as well as the traditionally trained CNN and the work of [Atoum et al. 2017] (which proposes a much complex CNN), better than approaches that require complex and expensive architectures. Besides, when compared with the traditionally trained CNN, lsCNN training was again much faster (lsCNN obtained its best performance on iteration $9,800$ against iteration $80,900$ for the lsCNN architecture traditionally trained, as mentioned).

1) Connection Weights: Still referring to the intra-databases experiments on the Replay-Attack [Chingovska et al. 2012] and CASIA [Zhang et al. 2012] datasets, in order to better analyze the behavior of the weights of the lsCNN model given its two-steps training, we verified the values of the convolutional kernels between the first and second layers of the proposed model (which represent weights of connections between neurons in such layers) after the global fine-tuning step of the proposed architecture. In subsection 12.3.1 of the thesis, one can see the configuration of the sinaptic weights of lsCNN and of a traditionally trained network. As one can observe, even after the global fine-tuning step in the training of lsCNN, the weights inherited from the PatchNets (main diagonal) remain of great magnitudes indicating that the local features are really important for the model (which preserved the high magnitudes of the locally learned weights). The traditionally trained CNN presented much more random weights after training, paying attention in much granular details (noise), not representative for face classification.

2) Statistical Analysis: Still regarding the intra-database experiments on the Replay-Attack [Chingovska et al. 2012] and CASIA [Zhang et al. 2012] datasets, For better comparison

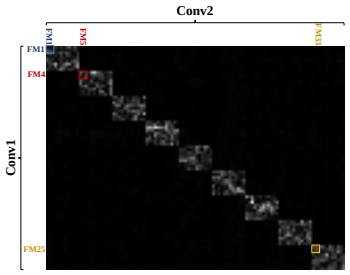

Fig. 6. Weights of convolutional kernels between layers $Conv1$ and $Conv2$ of lsCNN trained on Replay-Attack dataset after the global fine-tuning step.

between the lsCNN and the traditionally trained CNN, we repeated the training and test of lsCNN and of a traditional CNN on the Replay and CASIA datasets, 5 times each. In each experiment we measured the EER and HTER (the latter only for the Replay-Attack database) in all training iterations. We performed these experiments in order to compare the lsCNN and the traditional model in a more robust way. Fig. 7 shows the mean curves obtained by each model. The mean EER in both databases (Replay-Attack and CASIA) decreases faster and keep lower for the lsCNN model than for traditional CNN.

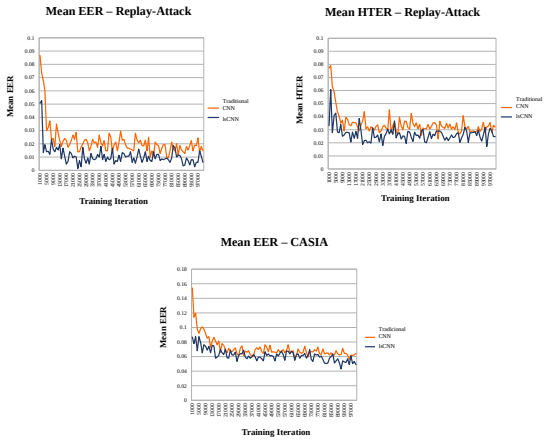

Fig. 7. Mean EER (and HTER) for the Replay-Attack and CASIA datasets.

V. CONCLUSION, PUBLICATIONS AND AWARDS

Based on the proposed approaches for efficient deep neural networks is possible to conclude that more efficient Deep Learning architectures, especially ones applied for face PA detection are feasible. The thesis with further details is available at: https://repositorio.ufscar.br/handle/ufscar/11609.

The following papers and awards were obtained related to the thesis:

Main publications

1) **"On the Learning of Deep Local Features for Robust Face Spoofing Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2018;
2) **"Partial Learning - On the Importance of Deep Local Features for Face Presentation Attack Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, (under submission);

3) **"Efficient Width-Extended Convolutional Neural Network for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, J. P. Papa e A. N. Marana, Anais da *Brazilian Conference on Intelligent Systems* (BRACIS), p. 230–235, 2018;
4) **"Deep Discriminative Restricted Boltzmann Machine (DDRBM) for Robust Face Spoofing Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, *Progr. in Human-Computer Int.*, v. 1, n. 3, p. 1–8, 2018;
5) **"Deep Texture Features for Robust Face Spoofing Detection"**, G. B. Souza, D. Santos, R. Pires, A. N. Marana e J. P. Papa, *IEEE Trans. on Circuits and Systems II*, v. 64, n. 12, p. 1397–1401, 2017 (abstract and full paper);
6) **"Efficient Transfer Learning for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Proc. of *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 643–651, 2017;
7) **"Detecção de Spoofing Facial: Uma abordagem baseada nas Máquinas de Boltzmann Restritas"**, G. B. Souza, A. N. Marana e J. P. Papa, *Revista Eletrônica de Matemática*, v. 10, p. 158-166, 2017;

Related publications

1) **"Cross-Domain Deep Face Matching for Real Banking Security Systems"**, J. S. Oliveira, G. B. Souza, A. R. Rocha, F. E. Deus e A. N. Marana, *Int. Conf. on e-Democracy and e-Government*, 2020.
2) **"Deep Features Extraction for Robust Fingerprint Spoofing Attack Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, *Journal of Artificial Intelligence and Soft Computing Research*, v. 9, n. 1, p. 41–49, 2018;
3) **"Introduction to Deep Learning - Theory and Practice"**, G. B. Souza, J. P. Papa e A. N. Marana, Anais da *Brazilian Conference on Intelligent Systems* (BRACIS), p. 1–2, 2018 (*abstract*);
4) **"Deep Boltzmann Machines for Robust Fingerprint Spoofing Attack Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Anais da *International Joint Conference on Neural Networks* (IJCNN), p. 1863–1870, 2017;
5) **"A 2D Deep Boltzmann Machine for Robust and Fast Vehicle Classification"**, D. F. S. Santos, G. B. Souza e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2017;
6) **"A Robust Restricted Boltzmann Machine for Binary Image Denoising"**, R. G. Pires, D. F. S. Santos, L. A. M. Pereira, G. B. Souza, A. L. M. Levada e J. P. Papa, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2017;
7) **"A Deep Boltzmann Machine-Based Approach for Robust Image Denoising"**, R. G. Pires, D. F. S. Santos, G. B. Souza, A. N. Marana, A. L. M. Levada e J. P. Papa, Anais do *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 525–533, 2017;
8) **"A Restricted Boltzmann Machine-Based Approach for Robust Dimensionality Reduction"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Proc. of *Workshop de Visão Computacional* (WVC), p. 138–143, 2017;
9) **"A Graph-Based Approach for Contextual Image Segmentation"**, G. B. Souza, G. M. Alves, A. L. M. Levada, P. E. Cruvinel e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2016;
10) **"Shape Analysis Using Multiscale Hough Transform Statistics"**, L. A. Ramos, G. B. Souza e A. N. Marana, Anais do *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 452–459, 2015;

- **2016** - **2^{nd} place** - Face Recognition Challenge at the *International Summer School for Advanced Studies on Biometrics for Secure Authentication*, Alguero (Italy);
- **2016** - **Top Best Works** - I Workshop of the Graduate Program in Computer Science, Federal University of São Carlos (UFSCar);
- **2017** - ***IEEE CIS Student Travel Grant Award*** for presenting the paper at the *International Joint Conference on Neural Networks*, 2017.
- **2018** - **Tutorial on "*Deep Learning - Theory and Practice*"** - Collocated event at the *Brazilian Conference on Intelligent Systems* (BRACIS) 2018.

REFERENCES

[Ackley et al. 1985] Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169.

[Akhtar and Foresti 2016] Akhtar, Z. and Foresti, G. L. (2016). Face spoof attack recognition using discriminative image patches. *Journal of Electrical and Computer Engineering*, 16.

[Atoum et al. 2017] Atoum, Y., Liu, Y., Jourabloo, A., and Liu, X. (2017). Face anti-spoofing using patch and depth-based CNNs. In *Proceedings of International Joint Conference on Biometrics*.

[Boulkenafet et al. 2015] Boulkenafet, Z., Komulainen, J., and Hadid, A. (2015). Face anti-spoofing based on color texture analysis. In *Anais da International Conference on Image Processing*, pages 2636–2640. IEEE.

[Chingovska et al. 2012] Chingovska, I., Anjos, A., and Marcel, S. (2012). On the effectiveness of local binary patterns in face anti-spoofing. In *Anais da International Conference of Biometrics Special Interest Group (BIOSIG)*, pages 1–7.

[Cortes and Vapnik 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[Hinton 2002] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. In *Neural Computation*, pages 1771–1800.

[Hinton 2012] Hinton, G. E. (2012). A practical guide to training Restricted Boltzmann Machines. In Montavon, G., Orr, G. B., and Müller, K. R., editors, *Neural Networks: Tricks of the trade*. Springer, United States.

[Howard et al. 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861.

[ISO 2016] ISO (2016). ISO/IEC 30107 - Presentation attack detection.

[Jain et al. 2011] Jain, A. K., Ross, A., and Nandakumar, K. (2011). *Introduction to Biometrics*. Springer, United States.

[Jia et al. 2014] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R. B., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093.

[Kingma and Ba 2015] Kingma, D. and Ba, J. (2015). Adam: a method for stochastic optimization. In *Anais da International Conference for Learning Representations*.

[Krizhevsky et al. 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1106–1114.

[LeCun et al. 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

[Li et al. 2016] Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., and Hadid, A. (2016). An original face anti-spoofing approach using partial convolutional neural network. In *Proc. of International Conference on Image Processing Theory, Tools and Applications*, pages 1–6.

[Lucena et al. 2017] Lucena, O., Junior, A., Moia, V., Souza, R., Valle, E., and de Alencar Lotufo, R. (2017). Transfer learning using convolutional neural networks for face anti-spoofing. In *Anais da International Conference on Image Analysis and Recognition*, pages 27–34.

[Ojala et al. 2002] Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 971–987.

[Parkhi et al. 2015] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *Anais da British Machine Vision Conference*.

[Schwartz et al. 2011] Schwartz, W., Rocha, A., and Pedrini, H. (2011). Face spoofing detection through Partial Least Squares and Low-Level Descriptors. In *Int. Joint Conf. on Biometrics*, United States. IEEE.

[Souza et al. 2017] Souza, G. B., Santos, D. F. S., Pires, R. G., Marana, A. N., and Papa, J. P. (2017). Efficient transfer learning for robust face spoofing detection. In *Anais do Iberoamerican Congress on Pattern Recognition*.

[Tan et al. 2010] Tan, X., Li, Y., Liu, J., and Jiang, L. (2010). Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *Anais da European Conference on Computer Vision*, pages 504–517.

[Tang et al. 2012] Tang, Y., Salakhutdinov, R., and Hinton, G. E. (2012). Robust Boltzmann Machines for recognition and denoising. In *Proc. of Conference on Computer Vision and Pattern Recognition*, United States. IEEE.

[Wang et al. 2018] Wang, Z., Tang, X., Luo, W., and Gao, S. (2018). Face aging with identity-preserved conditional generative adversarial networks. In *Anais da Conference on Computer Vision and Pattern Recognition*.

[Yang et al. 2014] Yang, J., Lei, Z., and Li, S. Z. (2014). Learn Convolutional Neural Network for face anti-spoofing. *CoRR*, abs/1408.5601.

[Zhang et al. 2012] Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., and Li, S. (2012). A face antispoofing database with diverse attacks. In *Proc. of International Conference on Biometrics*, Unites States. IEEE.