ON THE ACCURACY AND ROBUSTNESS OF DEEP TRIPLET EMBEDDING FOR FINGERPRINT LIVENESS DETECTION

Federico Pala and Bir Bhanu

Center for Research in Intelligent Systems
University of California, Riverside, Riverside, CA 92521, USA

ABSTRACT

Liveness detection is an anti-spoofing technique for dealing with presentation attacks on biometrics authentication systems. Since biometrics are usually visible to everyone, they can be easily captured by a malignant user and replicated to steal someone's identity. In particular, fingerprints can be easily reproduced by using gummy materials and attached to the impostor's fingertips, making the attack go unnoticed by security personnel and camera networks. In this paper, the classical binary classification formulation (live/fake) is substituted by a deep metric learning framework that can generate a representation of real and artificial fingerprints and explicitly models the underlying factors that explain their inter- and intra-class variations. The framework is based on a deep triplet network architecture and consists of a variation of the original triplet loss function. Experiments show that the approach can perform liveness detection in real-time outperforming the state-of-the-art on several benchmark datasets.

Index Terms— Biometrics, Security, Fingerprint, Liveness Detection, Deep Learning

1. INTRODUCTION

Currently, anti-spoofing techniques are increasingly becoming critical for biometrics systems since a large number of people use these technologies to access their personal data and for security purposes such as passing the security checks at airports. Among all the weak points of an authentication system, the biometric scanner is probably the most vulnerable part since it is in direct contact with the potential malignant user that wants to log in. The fingerprint scanner is particularly easy to spoof since the pattern on the fingertips can be easily obtained from a high-resolution photograph or a print left on various surfaces such as a mug. The pattern can be reproduced on gummy materials such as silicone and gelatine, to build a replica that can be directly applied to the sensing device [1]. At this point, the attacker would be able to fool the verification system by declaring the identity of the real owner of the fingerprint. Liveness detection is a technique to prevent these so called presentation attacks [2], by formulating a binary classification problem to establish whether the biometrics under examination comes from the real fingertips or an artificial replica [3].

Liveness detection systems can be classified into hardware and software systems. Hardware systems [3] use additional information to spot the characteristics of a living human fingerprint (e.g., by measuring the pulse [4]). In this work, we propose a software system, which falls under those techniques that can be introduced into any sensor software development toolkit without requiring additional sensing devices.

Previous Work. A significant part of the software based liveness detection literature is based on the extraction of different textural characteristics from the fingerprint pattern, followed by a binary classifier trained to discriminate between real and fake samples. For instance, [5] points out that the perspiration pattern of the skin manifests itself into static and dynamic patterns on the dielectric mosaic structure of the skin. The classification is performed using a neural network that takes as input a set of measures extracted from the data.

Recently, different studies [6, 7, 8] took advantage of the recent breakthroughs of deep learning in perceptual tasks [9], and have shown their effectiveness for the task of fingerprint liveness detection. In this paper, we principally take into consideration the approaches of [6] and [8]. In [6] deep Siamese networks have been considered along with classical pre-trained convolutional networks. Siamese networks are used to learn a similarity metric between an enrollment fingerprint, that is supposed to be a genuine acquisition, and a real or fake sample coming from the same individual. A limitation of this framework consists of requiring the enrollment fingerprint to evaluate the liveness of any acquisition. This is not possible every time since the fingerprint signature could have already been extracted in terms of minutiae information and encapsulated into a document such as a biometric passport. In other cases, the fingerprint could have been extracted using a different sensor, and it is well known that currently, cross-sensor liveness estimation lacks in robustness [8]. In [8] different convolutional neural network architectures have been proposed. Their best performing model is a VGG [10] architecture, previously trained on the Imagenet dataset of natural images [11] and then finetuned on fingerprint datasets.

Main Contribution. In this paper, we propose a liveness detection framework that overcomes the limitations of the current deep learning approaches. In particular, thanks to a patch based representation, it does not require a very deep architecture, allowing for mobile and off-line implementations. Further, it does not require the enrollment fingerprint of the user being captured. Comparative experiments show that along with state-of-the-art performance, superior robustness is achieved on unseen spoof materials.

2. TECHNICAL APPROACH

In this section, we describe the proposed approach for fingerprint liveness detection based on triplet loss embedding. From a training set of real and fake fingerprints, we collect a random fixed sized patch from each image. The patches are then arranged in a certain number of triplets $\{x_i, x_j^+, x_k^-\}$, where x_i (anchor) and x_j^+ are two examples of the same class, and x_k^- comes from the other class. We alternatively set the anchor to be a real or a fake fingerprint patch.

The architecture is composed of three convolutional networks with shared weights, so that three patches can be processed at the same time and mapped into a common feature space. The single embedding network is inspired by [12] where max-pooling units, widely used for down-sampling purposes, are replaced by simple convolution layers with increased stride. Table 2 contains the list of the operations performed by each layer of the embedding networks.

We denote by $\mathbf{r}(\cdot)$ the representation of a given patch obtained from the output of one of the three networks. The deep features obtained from the live and fake fingerprints are compared in order to extract an intra-class distance $d(\mathbf{r}(x), \mathbf{r}(x^+))$ and an inter-class distance $d(\mathbf{r}(x), \mathbf{r}(x^-))$. The objective is to capture the cues that make two fingerprints both real or fake. The real ones come from different people and fingers, and their comparison is performed to find some characteristics that make them genuine. At the same time, fake fingerprints come from numerous subjects and can be built using several materials. The objective is to detect anomalies that characterize fingerprints coming from a fake replica, *without regard to the material they are made of.*

Table 1. The architecture of the proposed embedding network. In the first column, the convolution filter size (FC - fully connected layer) along with stride (convolution sampling steps) and the amount of zero added to the sides, feature mapping, eventual batch normalization (BN) and non-linearity (ReLU - Rectified Linear Unit or SoftMax).

Layer description	Output size
Input: 32x32 gray level image	1x32x32
5x5 conv, stride=1, $1 \rightarrow 64$ + BN + ReLU	64x28x28
3x3 conv, stride=2, Pad=1, $64 \rightarrow 64$	64x14x14
3x3 conv, stride=1, $64 \rightarrow 128$ + BN + ReLU	64x12x12
3x3 conv, stride 2, Pad=1, $128 \rightarrow 128$	128x6x6
3x3 conv, stride=1, $128 \rightarrow 256$ + BN + ReLU	256x4x4
3x3 conv, filters, stride=2, Pad=1, $256 \rightarrow 256$	256x2x2
FC 4x256 $\rightarrow$ 256 + Dropout $p = 0.4$ + ReLU	256
FC $256 \rightarrow 256$ + SoftMax	256

Given a set of triplets $\{x_i, x_j^+, x_k^-\}$, where x_i is the anchor, and x_j^+ and x_k^- are respectively two examples of the same and the other class, the objective of the original triplet loss [13] is to give a penalty if the following condition is violated:

$$d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) - d(\mathbf{r}(x_i), \mathbf{r}(x_k^-)) + 1 \leq 0 \qquad (1)$$

At the same time, we would like to have the examples of the same class as close as possible so that, when matching a new fingerprint against the reference patches of the same type, the distance $d(\mathbf{r}(x_i), \mathbf{r}(x_j^+))$ is as small as possible. If we denote by $y(x_i)$ the class of a generic patch x_i, we can obtain the desired behavior by formulating the following loss function:

$$L = \sum_{i,j,k} \Big\{ c(x_i, x_j^+, x_k^-) + \beta c(x_i, x_j^+) \Big\} + \lambda \|\theta\|_2 \qquad (2)$$

where θ is a one-dimensional vector containing all the trainable parameters of the network, $y(x_i) = y(x_j)$, $y(x_k^-) \neq y(x_i)$ and:

$$c(x_i, x_j^+, x_k^-) = \big| d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) - d(\mathbf{r}(x_i), \mathbf{r}(x_k^-)) + 1 \big|_+ \qquad (3a)$$

$$c(x_i, x_j^+) = d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) \qquad (3b)$$

where $c(x_i, x_j^+, x_k^-)$ is the inter-class and $c(x_i, x_j^+)$ the intra-class distance term. $\lambda\|\theta\|_2$ is an additional weight decay term added to the loss function for regularization purposes. During training, we compute the subgradients and use backpropagation through the network to get the desired representation.

After a certain number of iterations k, we periodically generate a new set of triplets by extracting a different patch from each training fingerprint. It is essential to avoid updating the triplets after too many iterations because it can result in overfitting. At the same time, generating new triplets too often or mining hard examples can cause convergence issues.

For testing the liveness of a new fingerprint, any distance among bag of features such as the Hausdorff distance, can be used in order to match the query fingerprint $Q = \{\mathbf{r}(Q_1), \mathbf{r}(Q_2), \ldots, \mathbf{r}(Q_p)\}$ against the reference sets R_L and R_F. Since the training objective drastically pushes the distances to be very close to zero or one, a decision on the liveness can be made by setting a single threshold $\tau = 0.5$. It is also faster since it does not involve sorting out the distances.

Given a fingerprint Q, for each patch Q_j we count how many distances for each reference set are below the given threshold:

$$D(R_L, Q_j) = |\{i \in \{1, \ldots, n\} : d(R_{L_i}, Q_j) < \tau\}| \qquad (4a)$$

$$D(R_F, Q_j) = |\{i \in \{1, \ldots, n\} : d(R_{F_i}, Q_j) < \tau\}| \qquad (4b)$$

Then, we make the decision evaluating how many patches belong to the real or the fake class:

$$y(Q) = \begin{cases} \text{real} & \text{if } \sum_{j=1}^{p} D(R_L, Q_j) \geq \sum_{j=1}^{p} D(R_F, Q_j) \\ \text{fake} & \text{otherwise} \end{cases} \tag{5}$$

The above method can also be applied in scenarios where multiple fingerprints are acquired from the same individual, as usually happens on passport checks at airports. For instance, the patches coming from different fingers can be accumulated to apply the same majority rule of Eq. (5), or the decision can be made on the most suspicious fingerprint. See Figure 1 for a comparison between our framework and the Siamese approach proposed by [6].

3. EXPERIMENTS

We evaluate the proposed approach (TNet) with ten of the most popular benchmark for fingerprint liveness detection. We compare our method with state-of-the-art methods, specifically the pre-trained networks of [6] and [8], the Local Contrast Phase Descriptor LCPD [14] and the dense Scale Invariant Descriptor SID [15]. We strictly follow the competition rules using the training/test splits provided by the organizers. To establish the robustness to unseen materials, we instead follow the setup of [6].

Datasets. The LivDet datasets [16, 17, 18] were released since the first international fingerprint liveness detection competition, with the aim of becoming a reference and allowing researchers to compare the performance of their algorithms or systems. The organizers released several datasets, acquired using different fingerprint sensors and using different materials to produce the counterfeit replicas. Both LivDet 2009 and 2011 datasets have been obtained using the cooperative method, simulating the case where the victim voluntarily puts his/her finger on some moldable material. In the LivDet 2013 [18] competition instead, two datasets, Biometrika and Italdata, have been acquired using the non-cooperative method. That is, latent fingerprints have been collected from a surface, and then printed on a circuit board to generate a three-dimensional structure of the fingerprint that can be used to build a mold.

The size of the images, the scanner resolution, the number of subjects/samples and the materials used to fill the molds can be found on the reports published by the organizer of the competition [16, 17, 18].

Preprocessing. Since the images coming from the scanners contain a wide white area surrounding the fingerprint, we segment the images to avoid extracting background patches. We employ a simple classification rule based on the variance of the pixels. To exclude background noise that can interfere with the segmentation, we compute the connected components of the foreground mask and take the fingerprint region as the one with the largest area. To get a smooth segmentation, we generate the convex hull image from the binary mask using morphological operations.

Experimental Setup. For all the experiments, we evaluate the performance in terms of Average Classification Error (ACE). It is the average of the current standardized ISO/IEC 30107 metrics: the Normal Presentation Classification Error Rate (NPCER) and the Attack Presentation Classification Error Rate (APCER). Since the competition did not include a validation set, we reserved a fixed amount of 120 fingerprints for each dataset.

The triplets set for training is generated by taking one patch from each fingerprint and arranging them alternatively in two examples of one class and one of the other class. The set is updated every $k = 100,000$ triplets that are fed to the networks in batches of 100. We use stochastic gradient descent to minimize the triplet loss function, setting a learning rate of 0.5 and a momentum of 0.9. The learning rate is annealed so that after ten epochs it is reduced by half. The weight decay term of Eq. (2) is set to $\lambda = 10^{-4}$ and $\beta = 0.002$ as in [19].

After each epoch, we check the validation error. Instead of using the same accuracy measured at test (the average classification error), we construct $100,000$ triplets using the validation set patches but taking as anchor the reference patches selected from the training set and used to match the test samples. The error consists of the number of violating triplets and reflects how much the reference set failed to classify patches never seen before. Instead of fixing the number of iterations, we employ early stopping based on the concept of patience [20]. Each time the validation error decrease, we save a snapshot of the network parameters, and if in 20 consecutive iterations the validation error has not diminished, we stop the training and evaluate the accuracy on the test set using the last saved snapshot.

Experimental Results. In this section, we present the performance of the proposed fingerprint liveness detection system in different scenarios.

In Table 2 we list the performance in terms of average classification error on the LivDet competition test sets. With

Table 2. Average Classification Error on the test datasets.

Dataset	TNet	VGG [8]	LCPD [14]	SID [15]
Biometrika'09	**0.71**	4.1	1	3.8
CrossMatch'09	1.57	**0.6**	3.4	3.3
Identix'09	**0.044**	0.2	1.3	0.7
Biometrika'11	5.15	5.2	**4.9**	5.8
Digital'11	**1.85**	3.2	4.7	2.0
Italdata'11	**5.1**	8	12.3	11.2
Sagem'11	**1.23**	1.7	3.2	4.2
Biometrika'13	**0.55**	1.8	1.2	2.5
Italdata'13	0.5	**0.4**	1.3	2.7
Swipe'13	**0.66**	3.7	4.7	9.3
Average	**1.74%**	2.89%	3.8%	4.5%

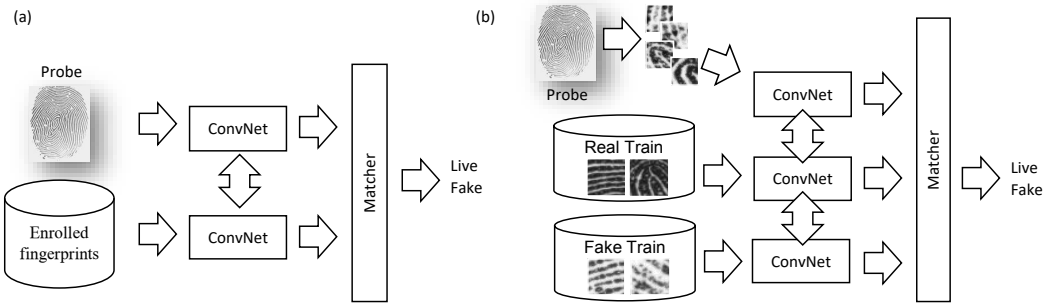

Fig. 1. Comparison between the (a) Siamese [6] and our (b) triplet architecture. While the Siamese network requires the enrollment database, our method evaluates the liveness by comparison with the same fingerprint set of patches used for training.

Table 3. Robustness to unseen materials on Biometrika 2013.

Training	Testing	TNet			Siamese [6]			GoogLeNet [6]		
		ACE	APCER	NPCER	ACE	APCER	NPCER	ACE	APCER	NPCER
All	All	0.55%	0%	1.1%	6.95%	6.1%	7.8%	3.4%	4.3%	2.5%
All	Ecoflex	0%	0%	0%	6.35%	4.9%	7.8%	1.25%	0%	2.5%
All	Gelatin	0%	0%	0%	13.85%	19.9%	7.8%	6.25	10%	2.5%
All	Latex	0.25%	0%	0.5%	4.95%	2.1%	7.8%	3.5%	4.5%	2.5%
All	Modasil	1%	0%	2%	4.85%	1.9%	7.8%	2.75%	3%	2.5%
All	WoodGlue	1.5%	0%	3%	6.85%	7.8%	5.9%	3.25%	4%	2.5%
All w/o Ecoflex	All	1.45%	2.4%	0.5%	5.2%	6.1%	4.3%	3.8%	3.2%	4.4%
All w/o Ecoflex	Ecoflex	0.3%	0%	0.6%	3.4%	2.5%	4.3%	2.2%	0%	4.4%
All w/o Gelatin	All	2.5%	4.8%	0.2%	4.7%	4.7%	4.7%	7.25%	7%	7.5%
All w/o Gelatin	Gelatin	2.5%	4.7%	0.3%	11.65%	18.3%	5%	11.6%	21%	2.2%
All w/o Latex	All	1.7%	2.7%	0.7%	12.6%	17.7%	7.5%	3.5%	5.5%	1.5%
All w/o Latex	Latex	1.35%	2%	0.7%	9.5%	11.5%	7.5%	4.8%	8%	1.6%
All w/o Modasil	All	1.6%	2.5%	0.7%	3.35%	3.9%	2.8%	3.7%	4.8%	2.6%
All w/o Modasil	Modasil	0.7%	0.5%	0.7%	1.4%	0%	2.8%	3.1%	3.5%	2.7%
All w/o WoodGlue	All	1.25%	1.8%	0.7%	7.25%	10%	4.5%	3.45%	3.4%	3.5%
All w/o WoodGlue	WoodGlue	0.55%	0.5%	0.6%	9.35%	14.2%	4.5%	2.5%	1.5%	3.5%

respect to the currently best-performing methods [8, 14, 15] we obtained competitive performance for all the datasets, especially for Italdata 2011, and Swipe 2013. Overall, our approach has an average error of 1.74% in comparison to the 2.89% of [8]. We point out that we did not use the dataset CrossMatch 2013 for evaluation purposes because the organizers of the competition found anomalies in the data and discouraged its use for comparative evaluations [21].

In Table 3 we show the performance in the cross-material scenario, simulating the case where a material unknown at training is given as a presentation attack. Also in this case, our approach is very competitive with respect to the pre-trained deep networks and the Siamese architecture of [6]. The spoofing materials easiest to detect are Ecoflex and Modasil with an average error of 0.3% and 0.6%, respectively. Gelatin is instead the most dangerous one since it presents an APCER of nearly 5%, way higher than the false negatives of other materials when hidden from training. Therefore, it is important to include gelatin samples to the training set to lower the classification error for all the materials.

The time to compute the deep representation from a single fingerprint, extracting 100 patches, is of 0.6ms using a single GPU and 0.3s using a Core i7-5930K 6 Core 3.5GHz desktop processor (single thread). The matching procedure takes 5.2ms on a single GPU and 14ms on the CPU. Finally, the training process converges after an average of 135 iterations. At each training iteration, the networks take 84s to handle $100,000$ triplets. On validation, since the weights are not updated, only 20s are required.

4. CONCLUSIONS

In this paper, we introduced a novel framework for fingerprint liveness detection which incorporates the recent advancements in deep metric learning. We validated the effectiveness of our approach in a scenario where the test fingerprints are acquired using the same sensing devices used for training. We also evaluated the robustness against unseen spoof materials for a particular sensor device. We obtained competitive or better performance for all the datasets. The method is found to be suitable for practical applications.

5. ACKNOWLEDGMENT

This work was supported in part by NSF grant 1330110 and ONR grant N00014-12-1-1026 . The contents of the information do not reflect the position or policy of US Government.

6. REFERENCES

[1] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial "gummy" fingers on fingerprint systems," in *Proceedings, Conference on Optical Security and Counterfeit Deterrence Techniques*, 2002, vol. 4677, pp. 275–289.

[2] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems: a survey," *IET Biometrics*, vol. 3, no. 4, pp. 219–233, 2014.

[3] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[4] K. Seifried, "Biometrics-what you need to know," *Security Portal*, vol. 10, 2001.

[5] R. Derakhshani, S. A. C. Schuckers, L. A. Hornak, and L. O'Gorman, "Determination of vitality from a noninvasive biomedical measurement for use in fingerprint scanners," *Pattern Recognition*, vol. 36, no. 2, pp. 383–396, 2003.

[6] E. Marasco, P. Wild, and B. Cukic, "Robust and interoperable fingerprint spoof detection via convolutional neural networks (hst 2016)," in *Proceedings, IEEE Symposium on Technologies for Homeland Security*, 2016, pp. 1–6.

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

[8] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206–1213, 2016.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Olga R., Jia D., Hao S., Jonathan K., Sanjeev S., Sean M., Zhiheng H., Andrej K., Aditya K., Michael B., Alexander C. B., and Li F., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[13] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proceedings, International Workshop on Similarity-Based Pattern Recognition (SIMBAD 2015)*. Springer, 2015, pp. 84–92.

[14] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Local contrast phase descriptor for fingerprint liveness detection," *Pattern Recognition*, vol. 48, no. 4, pp. 1050–1058, 2015.

[15] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, 2015.

[16] G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimberg, A. Congiu, A. Tidu, F. Roli, and S. Schuckers, "First international fingerprint liveness detection competition — livdet 2009," in *Proceedings, International Conference on Image Analysis and Processing (ICIAP 2009)*. 2009, pp. 12–23, Springer Berlin Heidelberg.

[17] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, "Livdet 2011 - fingerprint liveness detection competition 2011," in *Proceedings, IAPR International Conference on Biometrics (ICB 2011)*, 2012, pp. 208–215.

[18] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckcrs, "Livdet 2013 fingerprint liveness detection competition 2013," in *Proceedings, International Conference on Biometrics (ICB 2013)*, 2013, pp. 1–6.

[19] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016.

[20] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, pp. 437–478. Springer, 2012.

[21] L. Ghiani, D. A. Yambay, V. Mura, G. L. Marcialis, F. Roli, and S. A. Schuckers, "Review of the fingerprint liveness detection (livdet) competition series: 2009 to 2015," *Image and Vision Computing*, 2016 (in press).