Iris Liveness Detection by Relative Distance Comparisons

Federico Pala, Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, Riverside, CA 92521, USA
fedpala@ucr.edu, bhanu@cris.ucr.edu

Abstract

The focus of this paper is on presentation attack detection for the iris biometrics, which measures the pattern within the colored concentric circle of the subjects' eyes, to authenticate an individual to a generic user verification system. Unlike previous deep learning methods that use single convolutional neural network architectures, this paper develops a framework built upon triplet convolutional networks that takes as input two real iris patches and a fake patch or two fake patches and a genuine patch. The aim is to increase the number of training samples and to generate a representation that separates the real from the fake iris patches. The smaller architecture provides a way to do early stopping based on the liveness of single patches rather than the whole image. The matching is performed by computing the distance with respect to a reference set of real and fake examples. The proposed approach allows for real-time processing using a smaller network and provides equal or better than state-of-the-art performance on three benchmark datasets of photo-based and contact lens presentation attacks.

1. Introduction

Liveness detection is a preventive approach for containing sensor level attacks in biometrics authentication systems, where a malignant user builds a fake replica of a legitimate biometrics, applies it directly to the sensor and declares its corresponding identity. This task is formulated as a binary classification problem to establish if the claimed identity is genuine or it does not correspond to the subject in front of the sensor.

Currently, Presentation Attack Detection (PAD) techniques are increasingly becoming critical for biometrics systems since a large number of people use these technologies to access their personal data and for safety purposes such as passing the security checks at airports. Unfortunately, this massive usage of biometrics comes with various security and privacy issues. Different attacks can be directed to the authentication system to grant access to some exclusive area or to steal confidential data. For instance, the software and the network configuration can have security holes or bugs, and the matching algorithms can be fooled if the attacker knows the software implementation details. Moreover, whereas a physical key or badge can be replaced, the biometrics are permanent and their pattern, if visible, can be easily captured and reproduced.

Among all the weak points of an authentication system, the biometrics scanner is probably the most vulnerable part since it is in direct contact with the potential malignant user that has to be captured. Liveness detection is a technique to prevent these so called presentation attacks by formulating a binary classification problem to establish whether the biometrics under examination comes from a legitimate user or it is an illegitimate authentication attempt [9].

In this paper we focus on the iris biometrics [33, 35], where the pattern in the eyes can be easily obtained from a high-resolution photograph and then showed to a sensing device, fooling the authentication system by declaring the identity of the real biometrics owner (e.g., using a printed photo, a video on a tablet or printed contact lens). Figure 1 shows some examples of photos that simulate a photo-based presentation attack [5].

Figure 1. Samples from the Iris-2013-Warsaw dataset [5]. In the first column two real samples, followed by the corresponding print attacks.

Liveness detection systems can be distinguished as hardware and software systems. Hardware systems [9] use additional sensing devices to spot the characteristics of a living

2160-7516/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPRW.2017.95

human eye (e.g., the movement of the eyeball and 3D information). In this work, we propose a software system, which falls under those techniques that can be introduced into any sensor software development toolkit (SDK) without requiring additional sensing devices. The software of an iris scanner can be updated with no additional cost, and if a software technique is robust to a variety of attacks and does not annoy the users with too many false positives, it can be an alternative with respect to acquiring new sensing devices.

To date, few studies [21] took advantage of deep learning for the task of iris liveness detection. Deep learning has seriously improved the state-of-the-art in visual perception [18, 19, 10] and the ability to generate hierarchical representations and discover complex structures in raw images allows for better representations with respect to the traditional methods based on handcrafted features. Software-based systems for iris liveness detection can take advantage of the broad literature where similar tasks have already been addressed. In particular, we notice that a notion of similarity among real and fake iris samples has not yet been modeled. We make a step in this direction by proposing a deep metric learning approach based on Triplet networks [14]. Specifically, the three networks map the iris images into a representation space, where the learned distance captures the similarity of the examples coming from the same class and push away the real samples from the fake ones. Unlike other metric learning approaches, such as the ones involving Siamese networks [2], the triplet objective function puts in direct comparison the relations among the classes, giving a notion of context that does not require setting a threshold to make a new prediction [14]. Another advantage is the number of examples that can be generated by arranging the images in triplets. This is particularly useful in those biometrics applications where the amount of training data is not as large as more general artificial intelligence oriented datasets [26].

We propose a framework that learns a representation from iris patches, starting from a set of real and fake samples given as a training/reference set. Since at test time only an iris image is given, we make our decision on the basis of a matching score, computed against the reference set. The similarity metric is learned using a variation of the original triplet objective formulation [36, 14] that adds a pairwise term enforcing the vicinity of two samples of the same class [3]. We performed experiments for two kinds of presentation attacks: photo-based attacks via the Iris-2013-Warsaw dataset [5] employed for the LivDet[1] competition held in 2013 [38] and cosmetic lens attacks via the IIIT Cogent and Vista datasets [37]. The former simulates a malignant user showing to the sensor a photo of the victim eye, while the latter simulates an attack where the user is wearing contact lens printed with the victim's iris pattern.

The paper is structured as follows. In section 2 we present some of the current approaches for designing iris liveness detection systems, and the state-of-the-art. In section 3 we explain the details of the proposed framework and in section 4 we provide the experimental results. The final section 5 is dedicated to the conclusions.

2. Related Work and Contributions

In this section, we describe various iris liveness detection techniques, particularly with a focus on the ones related to our method, which can be considered as a static software based technique [9].

2.1. Hardware-based Systems

Hardware-based systems make use of additional sensing devices in order to test the liveness of the iris acquisition. The liveness detection is mainly performed in two different ways: by capturing some properties of the iris that are difficult or impossible to see from the iris scanner acquisition (e.g. tissue, blood vessels [35]) and by studying some behavioral characteristics. In the latter, we can further categorize methods that analyze the natural attitude of the eye (e.g. eye hippus, natural oscillation of the pupil [39]) and that study the response when solicited by an external stimulus (challenge/response, e.g. requiring the user to blink or looking into a particular direction/path [25]).

2.2. Software-based Systems

Software based techniques do not require additional hardware and can be integrated into the software development kit of the sensing device, taking advantage of the same image acquired for iris recognition. These techniques can be characterized by two kinds of characteristics: dynamic and static features [11]. Dynamic features involve behavioral characteristics such as eye blinking, pupil size oscillations and dynamic reflections that are particularly useful for photo-based attacks. One of the early works on iris liveness detection [7] pointed out that retinal light reflections commonly known as the red-eye effect can be a useful clue for detecting presentation attacks. Other approaches consist of controlling the light reflection from the cornea [5] and the pupil dynamics [6].

Static features, can also be extremely useful for iris liveness detection since, as firstly noticed by [7], printed images of irises present artificial artifacts that can be spotted by computer vision algorithms. He proposed a frequency analysis by applying a 2D Fourier transform in order to spot those frequencies that are probably the result of a printing attack. The work has been continued by other researchers, who applied different kinds of frequency analysis such as Wavelets [13] and Laplacian pyramids [24].

[1] http://livdet.org/

Other approaches consider instead the textural patterns of the iris image using popular texture descriptors such as Local Binary Patterns (LBP) [13], Binary Statistical Image Features (BSIF) [23], Scale-Invariant Feature Transform (SIFT) [20] and Local Phase Quantization (LPQ) [11]. For instance, in [29], SIFT features have been used to generate a hierarchical visual codebook that is able to generate a textural representation of irises for liveness detection.

2.3. Contributions of this Paper

This paper presents a framework for iris liveness detection that overcomes many of the problems with classical convolutional neural network architectures for biometrics applications.

First, an interesting point with respect to tasks such as generic image classification is the scarcity of data. Dealing with few samples is difficult since the networks are prone to overfitting. We use a smaller architecture and provide a way to do early stopping based on the liveness of single patches rather than the whole image. Classical early stopping algorithms would result in under fitting. With respect to [7] we introduced some of the recent advancements in deep metric learning and reduced the computational complexity. Thus, with respect to other iris liveness detection methods we introduced some improvements in the field.

Second, the computational complexity is very important since biometric authentication systems on mobile devices are implemented on a chip and the data acquisition is not visible by the software for security reasons. A small architecture proposed in our approach can be more easily implemented on the hardware.

Third, our approach extracts a signature ready to be matched using a simple Euclidean distance and the original image can be discarded which improves the security of the system.

Fourth, we show the results on three different datasets. We compare our results with respect to the Shift-Invariant Descriptor (SID) proposed in [17], refined in [16] and applied to liveness detection by [11]. We also compare with the only previous work (as the best of authors knowledge) using deep learning algorithms [22]. Further, we compare with other three handcrafted features based approaches, tested by [11] for liveness detection: the Dense SIFT descriptor, DAISY [30] and the Local Contrast-Phase Descriptor (LCPD) [12].

3. A Deep Triplet Embedding for Iris Liveness Detection

In this section we describe the proposed method for iris liveness detection based on triplet loss embedding. We start by describing the overall framework, subsequently we introduce the network architecture and the training algorithm. Finally, we describe the matching procedure that leads to the final decision on the liveness of a given iris image.

3.1. Framework

As depicted in Fig.2, the proposed framework requires a collection of real and fake iris images taken from a sensor and used as a training set. From each image, we randomly extract one fixed sized patch, and arrange them 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 iris patch.

The architecture is composed of three convolutional networks with shared weights, so that three images can be processed at the same time and mapped into a common feature space. 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 extracted from the live and fake iris acquisitions are compared in order to obtain 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 learn d so that the two examples of the same class are closer than two examples taken from different classes and two samples of the same class are as close as possible. After training the networks with a certain number of triplets, we extract a new patch from each training sample and generate a new set of triplets. This procedure is repeated until convergence, see more details in section 4.2.

After the training process is completed, the learned metric is used as a matching distance to establish the liveness of a new iris image. Given a query iris acquisition, we can extract p (possibly overlapping) patches and give them as input to the networks in order to get a representation $Q = \{\mathbf{r}(Q_1), \mathbf{r}(Q_2), \ldots, \mathbf{r}(Q_p)\}$. Since we are not directly mapping each patch to a binary liveness label, but generating a more fine-grained representation, the prediction can be made by a decision rule based on the learned metric d computed with respect to a set R_L and R_F of real and fake reference patches:

$$R_L = \{\mathbf{r}(x_{L_1}), \mathbf{r}(x_{L_2}), \ldots, \mathbf{r}(x_{L_n})\} \quad (1a)$$

$$R_F = \{\mathbf{r}(x_{F_1}), \mathbf{r}(x_{F_2}), \ldots, \mathbf{r}(x_{F_n})\} \quad (1b)$$

where the patches x_{L_i} and x_{F_i} can be taken from the training set or from a specially-made design set.

3.2. Network Architecture

We employ a network architecture inspired by [27] where max-pooling units, normally used for downsampling purposes, are replaced by convolution layers with increased stride. Table 1 contains the list of the operations performed by each layer of the embedding networks.

The architecture is composed of a first convolutional layer that takes the 32x32 grayscale iris patches and outputs 64 feature maps by using filters of size 5x5. Then,

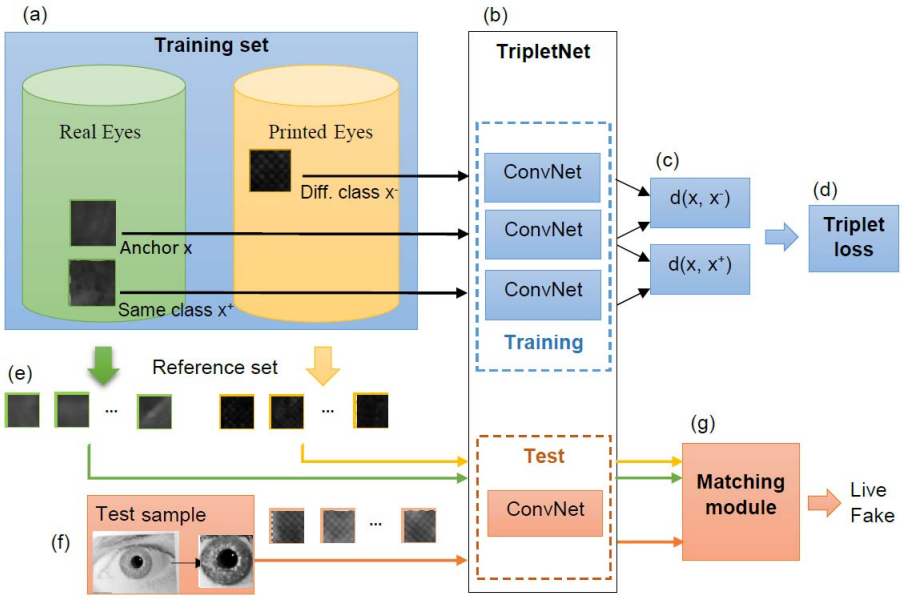

Figure 2. Overall architecture of the proposed iris liveness detection system. From the training set (a) of real and fake iris acquisitions, we train a triplet network (b) using alternatively two patches of one class and one patch of the other one. The output of each input patch is used to compute the inter- and intra-class distances (c) in order to compute the objective function (d) that is used to train the parameters of the networks. After training, a set of real and a set of fake reference patches (e) is extracted from the training set (two patches for each iris acquisitions) and the corresponding representation is computed forwarding them through the trained networks. At test time, a set of patches is extracted from the iris image (f) in order to map them to the same representation space as the reference gallery and are matched (g) in order to get a prediction on its liveness.

batch normalization [15] is applied to get a faster training convergence and rectified linear units (ReLU) are used as non-linearities. Another convolutional layer with a stride equal to 2, padding of 1 and filters of size 3x3 performs a downsampling operation by a factor of two in both directions.

The same structure is replicated two times, reducing the kernel size to 3x3 and increasing the number of feature maps from 64 to 128 and from 128 to 256. At this point, the feature maps have the size of 128x2x2 and are further processed by two fully connected layers with 256 outputs followed by a softmax layer. This non-linearity helps in getting a better convergence of the training algorithm and ensure that the distance among to outputs does not exceed one. Dropout [28] with probability 0.4 is applied to the first fully connected layer for regularization purposes.

The complete network is composed of three instances of this architecture: from three batches of iris images we get the L2 distances between the matching and mismatching images. At test, we take the output of one of the three networks to obtain the representation for a given patch. If there are memory limitations, an alternative consists of using just one network, collapse the three batches into a single one, and computing the distances among the examples corresponding to the training triplets.

Table 1. Architecture of the proposed embedding network.

Layer description	output
32x32 gray level image	
5x5 conv. filters, stride 1, 1 → 64 feat. maps	64x28x28
batch normalization	64x28x28
rectifier linear unit	64x28x28
3x3 conv. filters, stride 2, padding 1, 64 → 64 feat. maps	64x14x14
3x3 conv. filters, stride 1, 64 → 128 feat. maps	64x12x12
batch normalization	64x12x12
rectifier linear unit	64x12x12
3x3 conv. filters, stride 2, padding 1, 128 → 128 feat. maps	128x6x6
3x3 conv. filters, stride 1, 128 → 256 feat. maps	256x4x4
batch normalization	256x4x4
rectifier linear unit	256x4x4
3x3 conv. filters, stride 2, padding 1, 256 → 256 feat. maps	256x2x2
fully connected layer 4x256 → 256	256
dropout $p = 0.4$	256
rectifier linear unit	256
fully connected layer 256 → 256	256
softmax	256

3.3. Training

As schematized in Figure 3, the triplet architecture along with the triplet loss function aims to learn a metric that makes two patches of the same class closer with respect to two coming from different classes. The objective is to capture the cues that make two iris acquisitions both real or

fake. The real ones come from different people and eyes, and their comparison is performed in order to find some characteristics that make them genuine. At the same time, fake iris images come from different people and can be built using different techniques.

Given a set of triplets $\{x_i, x_j^+, x_k^-\}$, where x_i is the anchor and x_j^+ and x_k^- are two examples of the same and the other class, respectively, the objective of the original triplet loss [14] is to give a penalty if the following condition is violated:

$$d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) - d(\mathbf{r}(x_i), \mathbf{r}(x_k^-)) + 1 \leq 0 \quad (2)$$

At the same time, as proposed by [36, 3] we would like to have the examples of the same class as close as possible so that, when matching a new iris image 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:

$$L = \sum_{i,j,k} \left\{ c(x_i, x_j^+, x_k^-) + \beta c(x_i, x_j^+) \right\} + \lambda \|\theta\|_2 \quad (3)$$

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

$$c(x_i, x_j^+, x_k^-) = \left| d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) - d(\mathbf{r}(x_i), \mathbf{r}(x_k^-)) + 1 \right|_+ \quad (4a)$$

$$c(x_i, x_j^+) = d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) \quad (4b)$$

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 iris image. It is essential not to update 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.

3.4. Matching

In principle, any distance among bag of features such as the Hausdorff distance, can be used in order to match the query iris $Q = \{\mathbf{r}(Q_1), \mathbf{r}(Q_2), \ldots, \mathbf{r}(Q_p)\}$ against the reference sets R_L and R_F. Since the training objective drastically pushes the distances to be very close to zero or to one, a decision on the liveness can be made by setting a simple threshold $\tau = 0.5$. In particular, the Hausdorff distance would be too sensitive to outliers since it involves the computation of the minimum distance between a test patch and each patch of each reference set. Even if using the k-th Hausdorff distance [34], that considers the k-th value instead of the minimum, we obtained a better performance by following a simple majority voting strategy. It is also faster since it does not involve sorting out the distances.

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

$$D(R_L, Q_j) = |\{i \in \{1, \ldots, n\} : d(R_{L_i}, Q_j) < \tau\}| \quad (5a)$$

$$D(R_F, Q_j) = |\{i \in \{1, \ldots, n\} : d(R_{F_i}, Q_j) < \tau\}| \quad (5b)$$

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

$$y(Q) = \begin{cases} \text{real} & \text{if } \sum_{j=1}^{p} D(R_L, Q_j) \geq \sum_{j=1}^{p} D(R_F, Q_j) \\ \text{fake} & \text{otherwise} \end{cases} \quad (6)$$

The above method can also be applied in scenarios where both eyes are acquired from the same individual. For instance, the patches coming from different eyes can be accumulated in order to apply the same majority rule of Eq. 6, or the decision can be made on the most suspicious iris acquisition.

4. Experiments

In this section we evaluate the proposed approach for iris liveness detection on photo-based [5] and contact lens presentation attacks [37].

The network architecture along with the overall framework have been implemented using the Torch7 computing framework [4] on an NVIDIA® DIGITS™ DevBox with four TITAN X GPUs with 7 TFlops of single precision, 336.5 GB/s of memory bandwidth, and 12 GB of memory per board. MATLAB® has been used for image segmentation.

4.1. Datasets

We performed experiments on three datasets: the Iris-2013-Warsaw photo-based attacks dataset [5] and the IIIT Cogent and Vista cosmetic lens datasets [37].

The photo based attacks were acquired using a HP LaserJet 1320 and a Lexmark c534dn printer and a hole has been applied in place of the pupil check. Since the cameras usually search for pupil reflections, the simulated attacker would put the print over his/her eyes in order to fool the system. The number of live samples is 228 for training and 624 for testing while the number of fake acquisitions is 203 for training and 612 for testing. The number of distinct eyes is 284, while the number of distinct spoofed eyes is 276. More details are provided in the report of the competition [38].

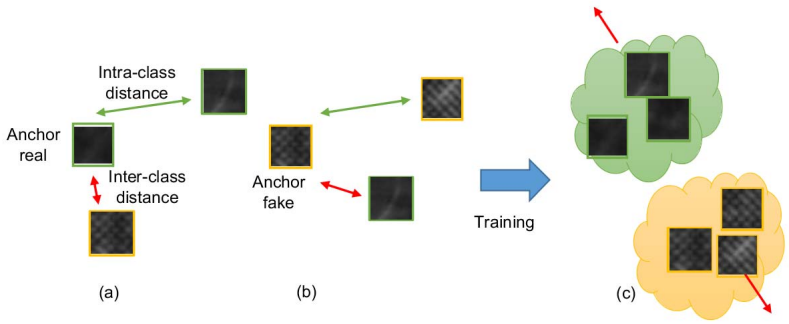

Figure 3. The training procedure uses examples as triplets formed by (a) two real irises (in green) and one impostor (in yellow) and (b) two impostors and one genuine. The training procedure using the triplet loss will result in an attraction for the irises of the same class (either real or fake) so that their distance will be as short as possible. At the same time real and fake irises will be pushed away from each other (c).

The IIIT-D cosmetic contact lens database [37] is composed of 6570 iris images appertaining to 101 individuals and acquired using two different iris sensors: Cogent and Vista. As in [11] we considered soft contact lens as live samples since they do not hide the iris pattern. Correct identification is instead not possible for the textured lens, therefore they are considered as a presentation attack.

4.2. Experimental Setup

For all the experiments we evaluate the performance in terms of Average Classification Error (ACE). This is the measure used to evaluate the entries in the LivDet competitions [38], and is the average of the following two ISO/IEC SC37 metrics: Attack Presentation Classification Error Rate (APCER) and the Bona Fide Presentation Classification Error Rate (BPCER). For all the experiments we follow the standard protocol and since a validation set has not been defined by the dataset providers, we reserved a fixed amount of 40 iris images from the training set (20 real and 20 fake).

For the Iris-2013-Warsaw photo-based attack dataset [5] we used the same training/test partition used for the competition. For the IIIT Cogent and Vista cosmetic lens datasets [37] instead, as in [11] we did a two-fold cross-validation using different subjects for each set.

The triplets set for training is generated by taking one patch from each iris acquisition 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 1000 batches of 100 examples. In the remainder of the paper we refer to each update as the start of a new iteration. 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 η_0 is annealed by following the form:

$$\eta = \frac{\eta_0}{1 + 10^{-4} \cdot b} \tag{7}$$

where b is the progressive number of batches that are being processed. That is, after ten iterations the learning rate is reduced by half. The weight decay term of Eq. 3 is set to $\lambda = 10^{-4}$.

After each iteration 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 taken from the training set and used to match the test samples. The error consists on the number of violating triplets, and reflects how much the reference patches failed to classify patches never seen before. Instead of fixing the number of iterations, we employ early stopping based on the concept of patience [1]. Each time the validation error decreases, we save a snapshot of the network parameters, and if in 5 consecutive iterations the validation error is not decreasing anymore, we stop the training and evaluate the accuracy on the test set using the last saved snapshot.

All the images were cropped to include only the iris region using the integro-differential operator of [8]. Then we normalized the resulting image with zero mean and unitary variance.

4.3. Experimental Results and Comparison

In this section we compare our framework against the SID descriptor [11], the convolutional neural network method [22], the dense SIFT Descriptor [11], the DAISY descriptor [30] and the LCPD descriptor [12]. In Table 2 we list the performance in terms of average classification error on the Iris-2013-Warsaw dataset [38] and the IIIT Cogent and Vista cosmetic lens datasets [37].

With respect to the current best performing methods [11, 21] we obtained a 0% error for the Iris-2013-Warsaw dataset, in line with the SID descriptor of [11]. For the Cogent and Vista datasets we get the lowest average classification error, especially, for Vista with an improvement of 72% with respect to the state-of-the-art [22].

Table 2. Average Classification Error for the three datasets: Iris-2013-Warsaw, IIIT Cogent and Vista. Different columns show results from different approaches: in column 2 our TripletNet based approach, in column 3 the SID descriptor [11], in column 3 the convolutional neural networks method [22], in column 4 the dense SIFT Descriptor [11] based approach, in column 5 the DAISY descriptor [30] and in column 6 the LCPD descriptor [12]. SID, Dense SIFT, DAISY and LCPD have been tested for iris liveness detection by [11].

Dataset	TripletNet	SID [11]	CNN [21]	Dense Sift [22]	DAISY [30]	LCPD [12]
Iris-2013-Warsaw	0.0	0.0	0.2	0.5	0.9	7.1
Cogent	5.5	6.2	-	13.9	17.2	11.0
Vista	0.7	3.5	-	2.5	8.8	3.1

4.3.1 Computational Efficiency

One of the key benefits of our approach is the computational time since the patch representation allows for scaling the matching procedure on different computational units.

The time to compute the deep representation from a single iris image, 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 36 iterations. At each training iteration, the networks take 84s to handle $100,000$ triplets. For validation, since the weights are not updated, only 20s are required.

5. Conclusions and Future Work

In this study, we introduced a framework for iris liveness detection which embeds the recent advancements in deep metric learning. We validated the effectiveness of our approach in scenarios where the iris acquisition system has been violated by photo-based and contact lens attacks. The approach is able to work in real-time and has a better accuracy over the state-of-the-art on two test benchmark datasets.

In conclusion, we point out that the employment of software based liveness detection systems should never give a sense of false security to their users. As in other areas such as cyber security, the attackers become more resourceful every day and new ways to fool a biometric system will be discovered. Therefore, such systems should be constantly updated and monitored, especially in critical application such as airport controls.

Future work will involve considering different kind of attacks, such as eyes extracted from cadavers [31]. Experiments will be performed on larger datasets considering subjects of different age, sex and ethnicity [32] that are acquired under different time periods, environments and using a variety of sensors with a multitude of spoofing attacks simulations.

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.

References

[1] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012. 6

[2] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993. 2

[3] 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. 2, 5

[4] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 5

[5] A. Czajka. Database of iris printouts and its application: Development of liveness detection method for iris recognition. In *2013 18th International Conference on Methods Models in Automation Robotics (MMAR)*, pages 28–33, Aug 2013. 1, 2, 5, 6

[6] A. Czajka. Pupil dynamics for iris liveness detection. *IEEE Transactions on Information Forensics and Security*, 10(4):726–735, 2015. 2

[7] J. Daugman. Iris recognition and anti-spoofing countermeasures. In *Proc. Int. Biometrics Conf., IBC*, 2004. 2

[8] J. G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1148–1161, 1993. 6

[9] J. Galbally, S. Marcel, and J. Fierrez. Biometric antispoofing methods: A survey in face recognition. *IEEE Access*, 2:1530–1552, 2014. 1, 2

[10] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 2

[11] 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*, 10(4):849–863, 2015. 2, 3, 6, 7

[12] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva. Local contrast phase descriptor for fingerprint liveness de-

tection. *Pattern Recognition*, 48(4):1050–1058, 2015. 3, 6, 7

[13] Z. He, Z. Sun, T. Tan, and Z. Wei. *Efficient Iris Spoof Detection via Boosted Local Binary Patterns*, pages 1080–1090. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 2, 3

[14] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *Proceedings, International Workshop on Similarity-Based Pattern Recognition (SIMBAD 2015)*, pages 84–92. Springer, 2015. 2, 5

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings, International Conference on Machine Learning (ICML 2015)*, 37, 2015. 4

[16] I. Kokkinos, M. Bronstein, and A. Yuille. *Dense scale invariant descriptors for images and surfaces*. PhD thesis, INRIA, 2012. 3

[17] I. Kokkinos and A. Yuille. Scale invariance without scale selection. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 3

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings, Conference on Advances in Neural Information Processing Systems (NIPS 2012)*, pages 1097–1105, 2012. 2

[19] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 2

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[21] 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*, 10(4):864–879, 2015. 2, 6, 7

[22] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falco, and A. Rocha. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, 10(4):864–879, April 2015. 3, 6, 7

[23] R. Raghavendra and C. Busch. Robust scheme for iris presentation attack detection using multiscale binarized statistical image features. *IEEE Transactions on Information Forensics and Security*, 10(4):703–715, April 2015. 3

[24] K. B. Raja, R. Raghavendra, and C. Busch. Presentation attack detection using laplacian decomposed frequency response for visible spectrum and near-infra-red iris systems. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, Sept 2015. 2

[25] C. Roberts. Biometric attack vectors and defences. *Computers & Security*, 26(1):14–25, 2007. 2

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[27] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 3

[28] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 4

[29] Z. Sun, H. Zhang, T. Tan, and J. Wang. Iris image classification based on hierarchical visual codebook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1120–1133, June 2014. 3

[30] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010. 3, 6, 7

[31] M. Trokielewicz, A. Czajka, and P. Maciejewicz. Human iris recognition in post-mortem subjects: Study and database. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–6. IEEE, 2016. 7

[32] M. Trokielewicz, A. Czajka, and P. Maciejewicz. Iris recognition under biologically troublesome conditions - effects of aging, diseases and post-mortem changes. In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOSIGNALS, (BIOSTEC 2017)*, pages 253–258, 2017. 7

[33] M. Vatsa, R. Singh, and A. Noore. Improving iris recognition performance using segmentation, quality enhancement, match score fusion, and indexing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):1021–1035, Aug 2008. 1

[34] J. Wang and J. Zucker. Solving multiple-instance problem: A lazy learning approach. In *Proceedings, International Conference on Machine Learning (ICML 2000)*, pages 1119–1126. Morgan Kaufmann, 2000. 5

[35] R. P. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997. 1, 2

[36] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3109–3118, 2015. 2, 5

[37] D. Yadav, N. Kohli, J. S. Doyle, R. Singh, M. Vatsa, and K. W. Bowyer. Unraveling the effect of textured contact lenses on iris recognition. *IEEE Transactions on Information Forensics and Security*, 9(5):851–862, May 2014. 2, 5, 6

[38] D. Yambay, J. S. Doyle, K. W. Bowyer, A. Czajka, and S. Schuckers. Livdet-iris 2013 - iris liveness detection competition 2013. In *IEEE International Joint Conference on Biometrics*, pages 1–8, Sept 2014. 2, 5, 6

[39] Y. Zhu, T. Tan, and Y. Wang. Biometric personal identification based on iris patterns. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 801–804. IEEE, 2000. 2