An Approach to Iris Contact Lens Detection based on Deep Image Representations

Pedro Silva, Eduardo Luz, Rafael Baeta, David Menotti
Computing Department
Federal University of Ouro Preto - UFOP
Ouro Preto, MG, Brazil, 35400-000
Email: {pedroh21.silva,eduluz,rafael.baeta,menottid}@gmail.com

Helio Pedrini, Alexandre Xavier Falcão
Institute of Computing
University of Campinas - UNICAMP
Campinas, SP, Brazil, 13083-852
Email: {helio,afalcao}@ic.unicamp.br

***Abstract*—Spoofing detection is a challenging task in biometric systems, when differentiating illegitimate users from genuine ones. Although iris scans are far more inclusive than fingerprints, and also more precise for person authentication, iris recognition systems are vulnerable to spoofing via textured cosmetic contact lenses. Iris spoofing detection is also referred to as liveness detection (binary classification of fake and real images). In this work, we focus on a three-class detection problem: images with textured (colored) contact lenses, soft contact lenses, and no lenses. Our approach uses a convolutional network to build a deep image representation and an additional fully-connected single layer with softmax regression for classification. Experiments are conducted in comparison with a state-of-the-art approach (SOTA) on two public iris image databases for contact lens detection: 2013 Notre Dame and IIIT-Delhi. Our approach can achieve a 30% performance gain over SOTA on the former database (from 80% to 86%) and comparable results on the latter. Since IIIT-Delhi does not provide segmented iris images and, differently from SOTA, our approach does not segment the iris yet, we conclude that these are very promising results.**

***Keywords*-Iris Biometrics; Contact Lens Detection; Deep Learning; Convolutional Networks.**

I. Introduction

Biometric-based person identification systems have been developed rapidly in the last two decades. Moreover, biometric systems based on iris recognition have been deployed in several applications, such as border-crossing control systems, controlled environments, access to personal computers and smart phones. Iris is considered as the most promising, reliable and accurate biometric trait, providing rich texture that allows high discrimination among subjects. Furthermore, iris is stable along ageing of individuals [1].

The first functional iris recognition method was introduced by Daugman in 1993 [1], whereas the first patent proposing iris texture as biometric modality appeared in 1987 [2]. Thenceforward, several iris recognition approaches have been proposed in the literature [3]–[5].

Due to the increasing use of iris as a source of biometric information in the last decade, the possibility of attacks to these systems has become more common [6]–[8]. These attacks are usually referred in the literature to as iris spoofing and several works for dealing with this problem have been proposed [9]–[11]. Nonetheless, the definition of iris spoofing detection may be confusing, where liveness and counterfeit detection terms are used with different meanings and, in some cases, interchangeably [12]. Several works in the literature have addressed the problem of classifying an iris image as real/live or as fake, in which a fake image is not a live one (e.g., a printed image [6], [10], [13]). In addition, counterfeit detection approaches have also been proposed in the past years [14]–[19], in which counterfeit iris with printed color contact lenses are considered fake images and iris images with soft/clear or no lenses are considered real images.

Given that cosmetic contact lenses are becoming more popular, the sort of attacks with textured contact lenses that an iris biometric system may suffer varies in a wide range. For instance, a person who has banned from a country or geographical region and has been included in a watch list may want to rejoin that region by using contact lenses to obfuscate his/her textured irises avoiding to be identified. Similarly, an individual may want to personify someone else by using textured contact lens iris of an enrolled person [9]. Moreover, transparent or prescript contact lenses used when acquiring iris images have indeed shown to decrease the false acceptance rate [15], [19] in iris recognition systems, demonstrating that it is important to identify when soft/clear lenses are present. Furthermore, the accuracy of textured contact lens detection methods may be affected by contact lens patterns and also sensor manufacturer as shown in [18].

In this context, we introduce the use of deep learning techniques [20]. In the last few years, deep learning has allowed promising and outstanding results for several and important visual analysis tasks, such as face recognition [21]–[24], pedestrian detection [25], character recognition [26], [27], traffic sign classification [28], general object recognition in large categorized databases [29], among others. Besides the success in these areas, the use of deep representations for spoofing detection on iris, face, and fingerprint images has also been recently proposed [10], in which a simpler two-class problem of detecting fake/spoof and real/live images is addressed.

The present paper addresses a more complex three-class image detection problem, where iris images may appear with textured (colored) contact lenses, soft contact lenses, and no lenses. We propose a convolutional network to build deep image representations, followed by a fully-connected single

layer with softmax regression for image classification. Our approach is based on the work of Krizhevsky et al. [29], in which the weights of all layers are learned by backpropagation.

In [30], the authors present two image databases to evaluate methods on the three-class detection problem: the 2013 Notre Dame Contact Lens Detection database (NDCL) and the IIIT-Delhi Contact Lens Iris database (IIIT-D). Each database contains images from two different sensors: LG4000 and AD100 in the NDCL database, where images come with iris location, and Cogent and Vista in the IIIT-D database, where iris location is not available (i.e., more challenging). We compare our approach with the state-of-the-art algorithm (SOTA), also proposed in [30], by taking into account images from each sensor and from different sensors.

The paper is organized as follows. In Section II, we present a brief review of relevant works directly related to contact lens spoofing detection. In Section III, the databases used in our experiments are described. The methodology proposed to cope with spoofing detection is detailed in Section IV. Experimental results are described and discussed in Section V. Finally, conclusions and directions for future work are outlined in Section VI.

II. Related Work

In this section, we review some relevant works directly related to the three-class iris image problem addressed in this paper, that is, those that propose to classify iris images in (color) textured contact lenses, soft (prescript or clear) contact lenses and non lenses.

The first step of a recognition system is to capture the iris images. Due to the difficulty in identifying the iris textures in the color images, sensors have to operate under near infrared (NIR) illumination. However, cosmetic contact lenses can change the pattern of the iris and its presence could be very difficult to detect on images taken under NIR illumination. Such undesirable feature is used against iris recognition systems, which makes spoofing attacks with textured lenses easier and also increases false negative matching even for prescript lenses [15], [17], [31].

Lee et al. [32] propose a new method for detecting fake iris based on the Purkinje image. To acquire the data, a conventional USB camera is used with modified CCD sensor and special illumination. To accomplish the experiments, the dataset is built with 300 live irises and 15 fake ones. The authors report a false accept rate (FAR) of 0.33% and false reject rate (FRR) of 0.33% on the dataset, however, a more robust evaluation, on a larger and diverse dataset, should be considered to properly validate the method.

Wei et al. [14] present three methods for detecting textured lenses: measurement of iris edge sharpness, application of iris-texton for characterizing the visual primitives of iris textures, and use of selected features based on co-occurrence matrix. To perform experiments, two datasets are built using CASIA [33] and BATH [34] datasets for live irises, whereas fake irises are collected by the authors. The reported correct classification rate (CCR) is up to 100% for experiments using co-occurrence matrix features.

In [35], a method based on Local Binary Pattern (LBP) encoding and Adaboost learning together with Gaussian kernel density estimation achieves FAR of 0.67% and FRR of 2.64% on discriminating fake iris texture from live iris. The method is evaluated on CASIA-Iris-V3 [36] and ICE v1.0 [37] with the addition of 600 custom fake iris images. Among the used 600 fake images, there are 20 different types of textured contact lenses.

In [16], it is proposed a contact lens detection algorithm based on Scale-Invariant Feature Transform (SIFT), weighted LBP and Support Vector Machines (SVM). According to the authors, the combination of SIFT and LBP improves its variance of scale illumination and local affine distortion. The authors claim that their method achieves state-of-the-art performance in contact lens detection. They build a custom dataset of 5000 fake iris images with 70 different types of textured lens.

After Daugman [38] presented a method for allowing easy detection of contact lens patterns, many other authors have reported accuracy rates over 98% [14], [16], [35]. However, since the contact lens technology is under constant development, robust detection has become more difficult [9]. Combined to this fact, some studies found in the literature are favored by their methodology due to the use of datasets containing contact lenses from a single manufacturer among both training and test data [9], [14]. According to [18], in more realistic scenarios, methods whose accuracy are close to 100% could decrease to below 60%.

To avoid this situation, two datasets are built in [18] with textured contact lenses from three major manufacturers. Multiple colors are selected for each manufacturer and some lenses are also designed to correct astigmatism. Authors claim that textured lens detection can drop dramatically when tested on a manufacturer of lenses not seen in the training data and when the iris sensor is different between training and test data. An extension of this work is presented in [30], where the datasets are well described and made available upon request. Additionally, state-of-the-art results are reported by a modified LBP feature extraction method and compared with 17 different classifiers. The databases are tested with techniques available in the literature, such as textural features based on co-occurrence matrix, weighted LBP approach, as well as other techniques based on LBP and SVM. Finally, the authors suggest that the development of a fully general approach to textured lens detection is a problem that still requires attention.

In a recent work [19], a new contact lens detection method based on binarized statistical image features reports accuracies near to optimality for the NDCL database. However, in that work, the authors deal with a two-problem classification, that is, soft/clear lens iris images are considered the same class as non lens iris images.

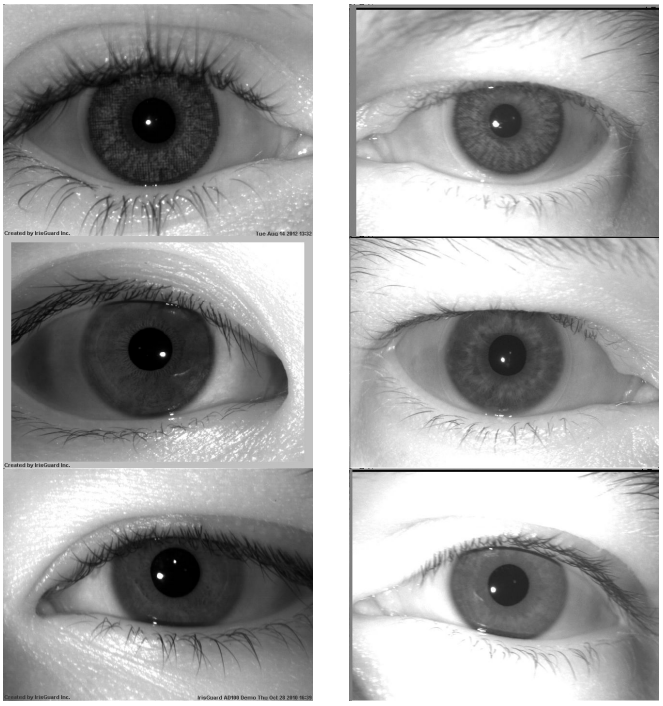

Fig. 1. Samples of images in the 2013 Notre Dame Contact Lens Detection (NDCL) database. In the first and second columns, we show images acquired with AD100 and LG4000 sensors, respectively. The first, second and third rows present samples with textured/cosmetic contact lenses, soft/clear/prescript contact lenses, and no contact lenses, respectively.

Fig. 2. Samples of images in the IIIT-Delhi Contact Lens Iris (IIIT). In the first and second columns, we show images acquired with Cogent and Vista sensors, respectively. The first, second and third rows present samples with textured/cosmetic contact lenses, soft/clear/prescript contact lenses, and no contact lenses, respectively.

III. Databases

In this section, we describe the databases used in our experiments. Both are publicly available upon request and were specifically developed for evaluation of contact lens iris detection in a three-class way [30]. We summarize the main characteristics of each database in Table I and present additional details in the following subsections. Note that all images of these databases are grayscale with 640×480 pixels.

A. Notre Dame Contact Lens Database

The 2013 Notre Dame Contact Lens Detection (NDCLD'13 or simply NDCL) database consists of 5100 images [39]. All 640×480 pixel images of this database were acquired under near-IR illumination using two types of cameras, LG4000 and IrisGuard AD100. This database is divided into two subsets: *LG4000* with 3000 images for training and 1200 for verification; *AD100* with 600 images for training and 300 for verification. These subsets are indeed used as primary databases for intra-camera evaluation.

The entire database, i.e., the fusion of images acquired by the LG4000 & AD100 cameras, is proposed as a multi-camera training set of 3600 images and a verification (testing) set of 1500 images. The images are equally divided into three classes: (1) wearing cosmetic contact lenses, (2) wearing clear soft contact lenses, and (3) wearing no contact lenses. Fig. 1 illustrates some samples of the NDCL and its cameras and classes.

All images in the database are annotated with the following information: an ID to the subject it belongs, eye (left and right), the subject's gender, race, the type of contact lenses used, and the coordinates of pupil and iris. These coordinates allow us to perform experiments considering perfect iris segmentation. More specific details for this database can be found in [39, Section II.B]

B. IIIT-D Contact Lens Iris Database

The Indraprastha Institute of Information Technology (IIIT)-Delhi Contact Lens Iris (IIIT-D CLI or simply IIIT) database contains 6583 iris images of 101 subjects. For each individual: (1) both left and right eyes were captured generating 202 iris classes (different iris); (2) images were captured without lens and with soft and textured lens - the three classes considered here; (3) the textured lenses were captured by using variations in iris sensors and lenses (colors and manufacturers). Images in this database are illustrated in Fig. 2.

The used iris sensors are *Cogent* dual and *Vista*FA2E single. Although this database offers a large variation of textured contact lenses, the iris location information is not provided.[1] More specific details for this database can be found in [39, Section II.A].

[1]We only conducted experiments on this database using the entire eye image. The perfect iris segmentation or annotation is planned as future work.

TABLE I
MAIN FEATURES OF THE DATASETS CONSIDERED HEREIN AND INTRODUCED IN [30].

Database	Sensor	# Training				# Testing/Verification				# Full			
		Text.	Soft	No	Total	Text.	Soft	No	Total	Text.	Soft	No	Total
NDCL	IrisGuard AD100	200	200	200	600	100	100	100	300	300	300	300	900
	LG4000 iris camera	1000	1000	1000	3000	400	400	400	1200	1400	1400	1400	4200
	Multi-camera	1200	1200	1200	3600	500	500	500	1500	1700	1700	1700	5100
IIIT	Cogent Scanner	589	569	563	1721	613	574	600	1787	1202	1143	1163	3508
	Vista Scanner	535	500	500	1535	530	510	500	1540	1065	1010	1000	3075
	Multi-scanner	1124	1069	1063	3256	1143	1084	1100	3327	2267	2153	2163	6583

IV. DEEP REPRESENTATIONS

In this section, we present the proposed method for iris contact lens detection based on deep image representations. Initially, we briefly describe the structure of the deep learning techniques used to build deep representations for the problem, which involves a combination of convolutional network [40], for deep image representations, and a fully-connected [41] three-layered network for classification. Then, we detail the methodology to choose the network topology and learn its parameters, by using the domain-knowledge from the literature. The activation operations used here are the rectified linear units (ReLU) [29], which have demonstrated to be essential to learn deep representations. Based on gain control mechanisms found in cortical neurons [42], the normalization operation promotes competition among filter outputs such that high and isolated responses are further emphasized [10]. Spatial pooling is a foundational operation in convolutional networks [40] that aims at bringing translational invariance to the features by aggregating activations from the same filter in a given region. The order of these last two operations, i.e., normalization and pooling, in a convolutional layer is an open problem and is application dependent. As we expect to achieve higher discrimination power with deep representations, the convolutional network stacks several layers for final image representation. All these operations and layers demand the determination of several parameters. Instead of performing random search on the hyperparameter space [24], [43] or even applying some specific search algorithm [44], we preferred to empirically analyze a set of parameters at a time to build the final network structure (topology), and to learn the filter weights by backpropagation. The idea of learning the network architecture by using random weights [10], [24], [43], [44] certainly deserves more attention and we will leave this approach for a future work. The idea here is to first evaluate how far one can go with the domain-knowledge from previous works for object classification [29], in CIFAR-10 database[2] and spoofing [10], to establish a preliminary topology network and explore its parameters according to the perception of the problem. These steps are explained in Section IV-A and employed in Section V.

The final layer of the convolutional network outputs a deep image representation. For classification, we use a fully-connected three-layered network [41]. We discard the use of unshared local layers, since the literature [10] has shown that they are inappropriate to problems in which the object structure is irrelevant. The last network contains only three neurons (one for each class) and classification is performed by softmax regression. Then, the weights of each layer in both networks are learned by the well-known backpropagation algorithm.

The framework described above appears in CUDA-convnet library implemented in C++ / CUDA by Krizhevky [3]. It is important to highlight that such networks are a longstanding approach, but it has recently enabled significant advances in computer vision and pattern recognition fields, due to the availability of more data and processing power, as well as a better understanding of the learning process [21], [29], [44].

A. Methodology

The development of a network architecture for the three-class detection problem, involving textured, soft, and no contact lens images, starts from the *Spoofnet* — a network specially developed to address the two-class detection problem of fake and live images [10]. From this network, we determine the range of the parameter values to evaluate and understand their influence on the performance of the contact lens detection method. These parameters are related to four groups: (i) the training methodology; (ii) the network architecture; (iii) the input image size; (iv) the database annotation. These groups are described in more details next.

Training methodology: We follow the training methodology established in [29] and described in [4]. An *initial learning rate* (LR) must be chosen. It is set to 10^{-3} in [29] and to 10^{-4} in [10]. We analyze both values.

Given an *initial number of epochs*, we develop the following steps in order to train a network: (1) train 100% of epochs in three out of four batches of the training data, using the fourth one as a validation set; (2) train more 40% of epochs in all four batches with the same learning rate; (3) train more 10% of epochs in all training batches by decreasing the LR by a factor of 10; (4) finally, train more 10% of epochs in all training batches by decreasing the LR again by a factor of 10. In [29], this initial number is set to 100, whereas in [10], it is set to 200. The authors in [10] argue that this parameter is both data and problem dependent. Then, here, besides evaluating 100 and

[2]http://www.cs.toronto.edu/~kriz/cifar.html

[3]https://code.google.com/p/cuda-convnet/

[4]https://code.google.com/p/cuda-convnet/wiki/Methodology

200 for the initial number of epochs, we propose to evaluate higher numbers while overfitting is not achieved. After those steps, we compute the accuracy of the trained network using the verification data.

Network architecture: Once the training methodology parameters are defined, we focus on the network topology definition. In the specification of the network architecture, one can use several layer and operation details [5], although here we evaluate: the *number of convolutional layers* of the networks: $\{1, 2, 3\}$, that is, networks with one, two, or three convolutional layers – the number of fully-connected layers is fixed in only one layer per class in order to reduce the number of possibilities to be evaluated; the *use or not of normalization operation* on top of each layer is also evaluated; the *number of filters* in each layer is also evaluated – combination of $\{16, 32, 64\}$ filters are evaluated for one, two and three layers. The window sizes of the convolutional, pooling and normalization operations are kept identical to the ones of *Spoofnet*.

Input image dimension: After finding the best network architecture, we investigate the influence of the input image size. We evaluate different *image sizes*, i.e., 64×64, 128×128 and 256×256 pixels, given that for lower values than 64×64, the contact lens details are not visible, whereas for higher values than 512×512, oversampling is performed and memory issues arise. To obtain images with the proposed dimensions, we resize them.

A very important aspect, that also affects the input image size, is data augmentation. It is strongly recommended to reduce overfitting. In Krizhevsky's framework [29], given an input image, it is possible to define a window size such that five image patches are cropped out from the original image. We define the border in pixels to be cropped out from the image. For instance, for a 64×64 input image, we consider the cropped image (a window) with 56×56 pixels at its center and we also slide this central window of 4 pixels horizontally and vertically to get cropped images from the four corners of the original image. We also apply reflections on each of the five images such that this procedure on each original image ends up to 10 training images. Here, we propose to evaluate crop border values of $\{2, 4, 6, 8\}$ for 64×64, $\{4, 8, 12, 16\}$ for 128×128, and $\{8, 16, 24, 32\}$ for 256×256 image sizes. Note that the crop border values respect a proportion regarding the image size.

Database annotation: As previously mentioned in Section III, the NDCL database (with images from the AD100 and LG4000 sensors) come with annotations for the pupil and iris locations, i.e., the x and y coordinates and the ratio, allowing a perfect iris segmentation or, in our case, only a perfect iris location, since we use squared region crops. For these datasets, through these annotations, we consider to use the iris image region plus a percentage of the background and define the following value: 0% (without), 10%, 20%, 30%, and 40%, in order to evaluate the importance of *background addition*.

[5]https://code.google.com/p/cuda-convnet/wiki/LayerParams

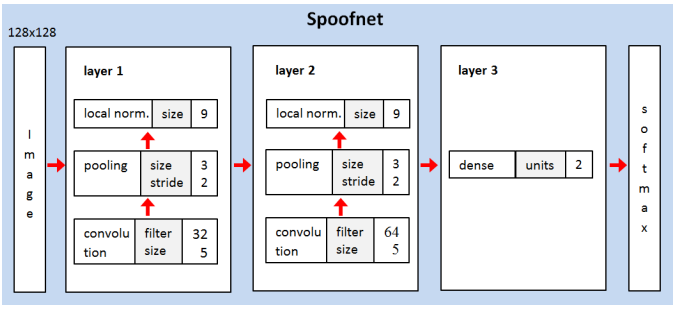

Fig. 3. *Spoofnet* - initial network topology used here. Source: [10].

V. Experiments and Results

In this section, we present the experiments performed in this work. We start by evaluating the groups of parameters established in the previous section to study their behavior and to obtain a performing network topology for contact lens detection called *CLDnet* (see Fig. 4). Then, we compare the effectiveness of our proposed approach with state-of-the-art results in different scenarios.

A. Parameter Evaluation

As established in Section IV-B, we have to first evaluate parameters in order to analyze their influence in the effectiveness of the proposed method and also design a robust network topology. These experiments were conducted separately only on NDCL database, namely AD100 and LG4000 sensors, since the iris location is available.

As initial network topology, we consider the one used in [10], i.e., *Spoofnet*. Its configurations are illustrated in Fig 3. We also consider image size of 128×128 and crop border of 8 pixels (indeed, input images are 112×112 pixels) – values used in *Spoofnet*. Furthermore, 10% of background addition was selected, before cropping for generating the initial input images. The 10% value was decided through visual inspection on the images and verifying that this amount suffices to include the contact lens borders in the cropped iris image.

The first evaluation is on the **training methodology**. We verified that, for initial learning rate of 10^{-3}, the framework caught/crashed in early iterations/epochs. This probably occurred because the learning rate was too aggressive. Then, for all remaining experiments, an *initial learning rate* of 10^{-4} was used. We start the *initial number of epochs* in 100, however, we also tested 200, 300 and 400 epochs. When 400 epochs were evaluated, we observed that the learning process was overfitting in the validation batch of the training set, then we decide to take 300 as the initial number of epochs, since the learning process was still achieving generalization. Therefore, this is defined as our evaluation protocol for the remaining experiments.

Then, we evaluate the **network architecture** parameters. We evaluate some configurations by varying the number of layers and the number of filters in each layer. The results in correct classification rate (CCR) are shown in Table II.

TABLE II
NETWORK ARCHITECTURE EVALUATION FOR AD100 AND LG4000 SENSORS ON NDCL DATABASE - VARYING THE NUMBER OF LAYERS AND THE NUMBER OF FILTERS IN EACH LAYER.

Sensor	N. Filters	CCR	N. Filters	CCR
AD100	16	72.33	16-16-16	73.67
	32	68.67	16-16-32	76.00
	64	70.00	16-16-64	77.00
	16-16	75.67	16-32-16	72.33
	16-32	75.00	16-32-32	76.33
	16-64	74.67	16-32-64	71.00
	32-32	76.00	32-32-16	75.00
	32-64	76.00	32-32-64	79.67
LG4000	16	79.50	16-16-16	77.59
	32	77.34	16-16-32	83.34
	64	80.84	16-16-64	81.17
	16-16	84.34	16-32-16	82.92
	16-32	84.82	16-32-32	81.75
	16-64	84.17	16-32-64	76.92
	32-32	85.59	32-32-16	81.34
	32-64	85.00	32-32-64	83.75

Note that the use of three convolution layers does not increase significantly the method effectiveness and the network using a single layer does not present promising results. The best result for the AD100 sensor (79.67%) is obtained using a three layers of convolutions, while only two layers yielded the best result for LG4000 sensor (85.59%). For our CLDnet, we kept the configuration of two layers using 32 and 64 filters for the first and second layers, respectively, since the results seemed more stable for both sensors of the NDCL database. We also evaluate whether or not to use the normalization operation, but the results demonstrated that the method effectiveness is insensitive to this operation in the contact lens detection problem. Thus, this operation was removed from the *CLDnet*.

Finally, we evaluate the **input image size** and **database annotations** parameters simultaneously. The results of these experiments are shown in Table III. By observing these results, we can conclude that, in general, the results achieved by the larger input image size, i.e., 256×256 pixels, correspond to the worst CCRs for both sensors, AD100 and LG4000. Additionally, on average, the results reported when using input image dimensions of 64×64 and 128×128 pixels are slightly

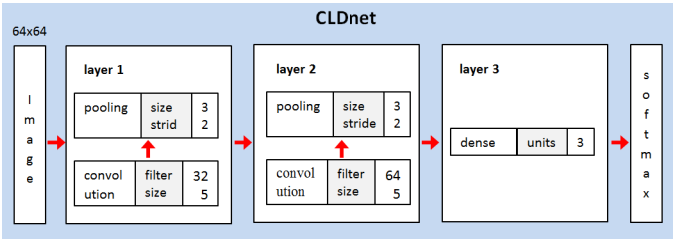

Fig. 4. *CLDnet* - network for Contact Lens Detection proposed here.

similar. As the image size might be a constraint in some applications, we prefer the smallest one for image input in our *CLDnet*. Moreover, the best results presented in Table III are obtained by networks with 64×64 pixel images as input, 4 pixels for crop border, and 10% of background addition, such that the final designed *CLDnet* shown in Fig. 4. Despite that fact, there is no strong claim to be stated for crop border and background addition parameters. That is, the results vary significantly when analyzing these two parameters.

B. Results

In this section, we compare the results obtained with our method against the state-of-the-art (SOTA) results in [30]. Tables IV, V and VI present CCRs for no (N), textured (T) and soft (S) contact lens classes and the overall (O) CCR when analyzing intra, inter, and multi-sensor evaluations, respectively. These results are analyzed as follows.

It is important to note that for the experiments run in the sensors of the IIIT-D database, we use the same network, *CLDnet*, however, we had to adjust the initial learning rate to 10^{-3}, because 10^{-4} was not sufficient for effective learning. All the remainder parameters and procedures were maintained as the sensors of the NDCL database.

1) Intra-sensor evaluation: It is possible to observe that the proposed method outperformed SOTA for AD100 & LG4000 sensors in the NDCL database, in which iris location is available, therefore, establishing new SOTA results. A marginal improvement is observed for the AD100 sensor images, however, for the LG4000 sensor, the CCR raises from approximately 80% to 86%, an improvement of 30%. We can

TABLE III
NETWORK ARCHITECTURE EVALUATION FOR AD100 AND LG4000 SENSORS ON NDCL DATABASE - EVALUATING THE INPUT IMAGE SIZE, THE CROP BORDER PARAMETER USED IN THE DATA AUGMENTATION, AND THE BACKGROUND ADDITION FROM THE DATABASE ANNOTATIONS.

Sensor	Background addition (%)	Input image size & Crop borders											
		64×64				128×128				256×256			
		2	4	6	8	4	8	12	16	8	16	24	32
AD100	0	74.67	73.67	71.00	71.00	74.00	78.00	70.67	70.00	70.33	70.67	71.33	63.33
	10	74.67	**78.33**	74.00	73.33	73.33	76.00	72.67	65.33	71.33	73.67	68.00	62.33
	20	71.33	76.67	76.00	67.33	69.67	75.33	76.33	68.00	71.33	71.00	73.00	68.33
	30	69.00	70.00	72.67	75.00	68.33	72.33	73.00	75.33	67.33	70.00	72.00	67.00
	40	66.33	69.67	72.67	69.67	73.33	71.67	71.00	68.33	66.67	69.67	75.67	68.33
LG4000	0	82.50	81.92	82.75	82.08	84.25	83.83	84.17	82.08	77.25	76.50	76.00	77.00
	10	83.25	**86.00**	84.25	82.75	84.58	84.58	85.25	82.92	72.25	75.58	76.25	75.08
	20	81.25	82.83	84.08	80.58	84.75	84.83	85.58	84.33	72.17	74.92	75.83	73.58
	30	82.00	81.92	82.50	80.00	83.42	82.83	84.33	84.42	71.08	70.42	74.50	71.33
	40	80.25	81.42	81.50	82.08	82.17	82.92	84.92	82.67	68.00	70.58	72.08	71.08

also see comparable results to SOTA for the Cogent & Vista sensor images in the IIIT-D database. In this case, iris location is not provided and the entire eye image was used as input to our method. Nonetheless, our method achieves higher results than the second best performing methods reported in [30]. The results on the IIIT-D database can be better understood, when we consider that SOTA counts with an iris segmentation algorithm.

TABLE IV
INTRA-SENSOR RESULTS FOR THE NDCL AND IIIT-D DATABASES.

	Sensors									
	AD100		LG4000		Cogent			Vista		
	Ours	SOTA	Ours	SOTA	Ours	SOTA	2nd	Ours	SOTA	2nd
N	73.00	81.00	84.50	76.21	35.50	66.83	59.73	60.80	76.21	49.49
T	97.00	100.00	99.75	91.62	73.00	94.91	91.87	55.88	91.62	99.42
S	65.00	52.00	73.75	67.52	98.21	56.66	52.84	98.30	67.52	59.32
O	**78.33**	77.67	**86.00**	80.04	69.05	73.01	68.57	72.08	80.04	69.84

2) Inter-sensor evaluation: Again, our method achieved new SOTA results in this scenario for the NDCL database, improving the CCR in 18% and 15%. In one sense, this result highlights how robust deep representations can be when learning features directly from the data. In contrast, disastrous results for the IIIT-D database were achieved due to the absence of iris location — a feature that comes with SOTA.

TABLE V
INTER-SENSOR RESULTS FOR THE NDCL AND IIIT-D DATABASES.

	Sensors							
Train	AD100		LG4000		Cogent		Vista	
Test	LG4000		AD100		Vista		Cogent	
	Ours	SOTA	Ours	SOTA	Ours	SOTA	Ours	SOTA
N	75.00	62.25	80.00	74.00	6.00	62.10	48.67	65.99
T	94.00	88.50	97.00	93.00	89.61	92.95	38.15	80.81
S	65.00	29.50	49.00	17.00	45.47	75.44	42.25	48.31
O	**78.00**	60.08	**75.33**	61.33	45.51	77.79	43.08	65.29

3) Multi-sensor evaluation: Finally, we observe that the CCRs obtained by our method outperforms the SOTA results in almost 10% in the multi-sensor scenario for the NDCL database and, even though the iris location is not provided (for the IIIT database), a comparable performance is achieved.

TABLE VI
MULTI-SENSOR RESULTS FOR THE NDCL AND IIIT-D DATABASES.

	Databases			
	NDCL		IIIT	
	Ours	SOTA	Ours	SOTA
N	77.40	72.60	47.55	62.14
T	99.60	97.00	61.07	94.74
S	71.40	50.00	97.99	61.63
O	**82.80**	73.20	69.28	72.96

C. Architecture learning and processing times

In our experiments, we used six PCs with 32GB RAM, Intel Core i7 CPUs, and NVIDIA GPUs (Tesla K40 with 12GB or GTX GeForce Titan Black with 6GB). The framework (CUDA-convnet) clearly relied on GPUs and the processing time of the different GPUs was not significant. The training time taken by the convolutional networks is highly dependent on the input image size, number of layers, and other parameters. For image sizes 256×256, 128×128, and 64×64, the average training time was less than 172, 49, and 11 minutes, respectively, for the LG4000 sensor – the one with the highest number of training samples.

Although we did not measure the classification time of a single sample, our approach is quite suitable for real-world applications. Indeed, there is an optimized framework, Jetpacs iOS Deep Belief image recognition framework [45], that implements the convolutional network architecture described in [29], which can classify a 256×256 image in one among 1k categories in less than 300ms on an iPhone 5S. That architecture is significantly larger and more complex than the ones that we propose here. Our architectures comprise fewer operations and layers, and use lower resolution images, 64×64 pixels. Therefore, contact lens detection systems with architectures developed by using [45] should be suitable for real-world applications.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the use of deep image representations, by means of learning weights in convolutional network followed by a classification network, for the iris contact lens detection problem. The conducted experiments validate our method, which could achieve a 30% performance gain over the state-of-the-art approach, SOTA, on the NDCL database and comparable results on the IIIT-D database. In NDCL, iris location is available, which allows to create deep image representations of regions of interest with mostly iris pixels. This becomes a problem in the IIIT-D database, where neither iris segmentation nor location is available. SOTA performs iris segmentation, but our approach is not prepared yet to preprocess images and segment/locate the iris. We intend to add this feature in future work and also to evaluate deep learning techniques in which the architecture of the network is first learned by using filters with random weights. Once the architecture is learned, the weights can be improved by backpropagation.

Effective comprehension and exploitation of representations built through deep learning techniques, such as the convolutional networks, are still open problems in the literature. We also plan to put more effort on this subject to clarify such points.

ACKNOWLEDGMENTS

We thank UFOP, Brazilian National Research Council – CNPq (Grants 307010/2014-7, 302970/2014-2, 479070/2013-0, 307113/2012-4), São Paulo Research Foundation – FAPESP, (Grants 2011/22749-8 and 2013/04172-0). D. Menotti thanks NVIDIA for donating two GeForce GTX Titan Black with 6GB each.

REFERENCES

[1] J. G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.

[2] L. Flom and A. Safir, "Iris Recognition System," U.S. US Patent 4 641 394, 1987.

[3] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn, "Image Understanding for Iris Biometrics: A Survey," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 281–307, 2008.

[4] Y. Song, W. Cao, and Z. He, "Robust Iris Recognition using Sparse Error Correction Model and Discriminative Dictionary Learning," *Neurocomputing*, vol. 137, pp. 198–204, 2014.

[5] A. F. M. Raffei, H. Asmuni, R. Hassan, and R. M. Othman, "Feature Extraction for Different Distances of Visible Reflection Iris using Multiscale Sparse Representation of Local Radon Transform," *Pattern Recognition*, vol. 46, no. 10, pp. 2622–2633, 2013.

[6] J. Galbally, S. Marcel, and J. Fierrez, "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition," *IEEE Trans. on Image Processing*, vol. 23, no. 2, pp. 710–724, 2014.

[7] P. Gupta, S. Behera, M. Vatsa, and R. Singh, "On Iris Spoofing using Print Attack," in *22nd Int. Conf. on Pattern Recognition*. IEEE, 2014, pp. 1681–1686.

[8] Z. Sun and T. Tan, "Iris Anti-Spoofing," in *Handbook of Biometric Anti-Spoofing*, ser. Advances in Computer Vision and Pattern Recognition, S. Marcel, M. S. Nixon, and S. Z. Li, Eds. Springer London, 2014, pp. 103–123.

[9] K. W. Bowyer and J. S. Doyle, "Cosmetic Contact Lenses and Iris Recognition Spoofing," *Computer*, vol. 47, no. 5, pp. 96–98, 2014.

[10] D. Menotti, G. Chiachia, A. Pinto, W. Schwartz, H. Pedrini, A. Falcão, and A. Rocha, "Deep Representations for Iris, Face, and Fingerprint Spoofing Detection," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, 2015.

[11] R. Raghavendra and C. Busch, "Robust Scheme for Iris Presentation Attack Detection Using Multiscale Binarized Statistical Image Features," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 4, pp. 703–715, 2015.

[12] Z. Sun, H. Zhang, T. Tan, and J. Wang, "Iris Image Classification Based on Hierarchical Visual Codebook," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1120–1133, Jun. 2014.

[13] A. Sequeira, H. Oliveira, J. Monteiro, J. Monteiro, and J. Cardoso, "MobILive 2014 - Mobile Iris Liveness Detection Competition," in *IEEE Int. Joint Conf. on Biometrics*, Sept 2014, pp. 1–6.

[14] Z. Wei, X. Qiu, Z. Sun, and T. Tan, "Counterfeit Iris Detection based on Texture Analysis," in *Int. Conf. on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[15] S. E. Baker, A. Hentz, K. W. Bowyer, and P. J. Flynn, "Degradation of Iris Recognition Performance due to non-Cosmetic Prescription Contact Lenses," *Computer Vision and Image Understanding*, vol. 114, no. 9, pp. 1030–1044, 2010.

[16] H. Zhang, Z. Sun, and T. Tan, "Contact Lens Detection based on Weighted LBP," in *Int. Conf. on Pattern Recognition*, 2010, pp. 4279–4282.

[17] N. Kohli, D. Yadav, M. Vatsa, and R. Singh, "Revisiting Iris Recognition with Color Cosmetic Contact Lenses," in *Int. Conf. on Biometrics*, 2013, pp. 1–7.

[18] J. S. Doyle, K. W. Bowyer, and P. J. Flynn, "Variation in Accuracy of Textured Contact Lens Detection based on Sensor and Lens Pattern," in *IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, 2013, pp. 1–7.

[19] J. Komulainen, A. Hadid, and M. Pietikainen, "Generalized Textured Contact Lens Detection by Extracting BSIF Description from Cartesian Iris Images," in *IEEE Int. Joint Conf. on Biometrics*, 2014, pp. 1–7.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013.

[21] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 132–142, to appear.

[22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[23] G. Chiachia, A. X. Falcão, N. Pinto, A. Rocha, and D. Cox, "Learning Person-Specific Representations From Faces in the Wild," *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 12, pp. 2089–2099, Dec. 2014.

[24] D. Cox and N. Pinto, "Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition," in *IEEE Int. Conf. on Automatic Face Gesture Recognition and Workshops*. IEEE, 2011, pp. 8–15.

[25] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian Detection with Unsupervised Multi-Stage Feature Learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3626–3633.

[26] D. Menotti, G. Chiachia, A. Falcão, and V. Oliveira Neto, "Vehicle License Plate Recognition with Random Convolutional Networks," in *27th SIBGRAPI Conf. on Graphics, Patterns and Images*, 2014, pp. 298–303.

[27] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep Big Simple Neural Nets For Handwritten Digit Recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[28] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-Column Deep Neural Network for Traffic Sign Classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[30] D. Yadav, N. Kohli, J. Doyle, R. Singh, M. Vatsa, and K. Bowyer, "Unraveling the Effect of Textured Contact Lenses on Iris Recognition," *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 5, pp. 851–862, 2014.

[31] S. E. Baker, A. Hentz, K. W. Bowyer, and P. J. Flynn, "Contact Lenses: Handle with Care for Iris Recognition," in *IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, 2009, pp. 1–8.

[32] E. C. Lee, K. R. Park, and J. Kim, "Fake Iris Detection by using Purkinje Image," in *Advances in Biometrics*. Springer, 2005, pp. 397–403.

[33] Chinese Academy of Sciences (CASIA), Institute of Automation, "CASIA Iris Image Database," http://biometrics.idealtest.org/findTotalDbByMode.do?mode=Iris, 2010, accessed 26 Mar, 2015 [online].

[34] University of Bath, Department of Electronic and Electrical Engineering, "University of Bath Iris Image Database," 2008.

[35] Z. He, Z. Sun, T. Tan, and Z. Wei, "Efficient Iris Spoof Detection via Boosted Local Binary Patterns," in *Advances in Biometrics*. Springer, 2009, pp. 1080–1090.

[36] Chinese Academy of Sciences (CASIA), Institute of Automation, "CASIA-IrisV3 Image Database [Online]," http://biometrics.idealtest.org/dbDetailForUser.do?id=3, 2010, accessed 26 Mar, 2015 [online].

[37] National Institute of Standards and Technology (NIST), "Iris Challenge Evaluation (ICE)," http://www.nist.gov/itl/iad/ig/ice.cfm, 2008, accessed 26 Mar, 2015 [online].

[38] J. Daugman, "Demodulation by Complex-Valued Wavelets for Stochastic Pattern Recognition," *Int. Journal of Wavelets, Multiresolution and Information Processing*, vol. 1, no. 1, pp. 1–17, 2003.

[39] J. Doyle and B. Kevin, "Notre Dame Image Database for Contact Lens Detection In Iris Recognition-2013: README," Available: http://www3.nd.edu/ cvrl/papers/CosCon2013README.pdf, 2014.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] C. Bishop, *Neural Networks for Pattern Recognition*, ser. Advanced Texts in Econometrics. Clarendon Press, 1995.

[42] W. S. Geisler and D. G. Albrecht, "Cortical Neurons: Isolation of Contrast Gain Control," *Vision Research*, vol. 32, no. 8, pp. 2429–2454, 1992.

[43] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A High-Throughput Screening Approach to Discovering Good Forms of Biologically-Inspired Visual Representation," *PLoS Computational Biology*, vol. 5, no. 11, 2009.

[44] J. Bergstra, D. Yamins, and D. D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," in *Int. Conf. on Machine Learning)*, 2013.

[45] P. Wardern, "The SDK for Jetpac's iOS Deep Belief image recognition framework," 2014. [Online]. Available: https://github.com/jetpacapp/DeepBeliefSDK