Convolutional Neural Networks for Iris Presentation Attack Detection: Toward Cross-Dataset and Cross-Sensor Generalization

Steven Hoffman, Renu Sharma, Arun Ross*
Department of Computer Science and Engineering
Michigan State University, USA
{hoffm470, sharma90, rossarun}@msu.edu

Abstract

Iris recognition systems are vulnerable to presentation attacks where an adversary employs artifacts such as 2D prints of the eye, plastic eyes, and cosmetic contact lenses to obfuscate their own identity or to spoof the identity of another subject. In this work, we design a Convolutional Neural Network (CNN) architecture for presentation attack detection, that is observed to have good cross-dataset generalization capability. The salient features of the proposed approach include: (a) the use of the pre-normalized iris rather than the normalized iris, thereby avoiding spatial information loss; (b) the tessellation of the iris region into overlapping patches to enable data augmentation as well as to learn features that are location agnostic; (c) fusion of information across patches to enhance detection accuracy; (d) incorporating a "segmentation mask" in order to automatically learn the relative importance of the pupil and iris regions; (e) generation of a "heat map" that displays patch-wise presentation attack information, thereby accounting for artifacts that may impact only a small portion of the iris region. Experiments confirm the efficacy of the proposed approach.

1. Introduction

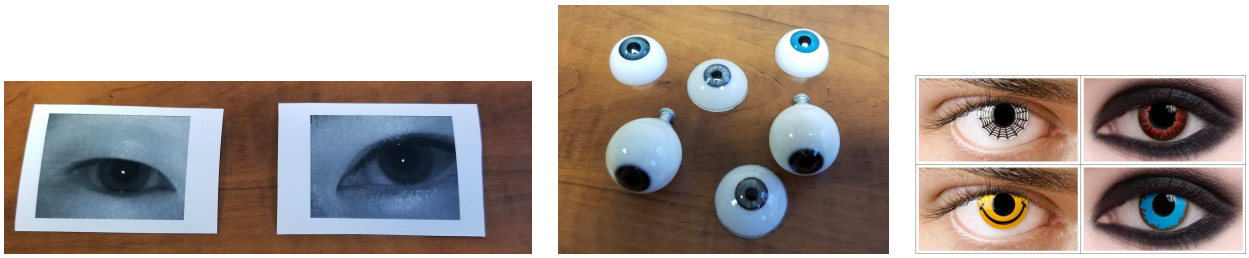
Iris biometric systems exploit the textural nuances of the iris in order to recognize individuals in an automated manner [26]. Despite their increasing popularity, iris recognition systems are vulnerable to *presentation attacks* [17, 3, 25]. A presentation attack (PA)[1] occurs when an adversarial user presents a fake or altered biometric sample to the sensor in order to spoof another user's identity, obfuscate their own identity, or create a virtual identity. Developing a robust and accurate presentation attack detection (PAD) module is, therefore, essential to maintain system integrity. Many possible attacks have been noted in the literature (Figure 1) based on printed iris images [15, 9], plastic, glass, or doll eyes [15, 28], cosmetic contact lenses [15, 9, 23], replaying a video of an individual's eye [3, 22], cadaver eyes [3, 19], robotic eyes [14], holographic eye images [3, 19], or coercing an individual to present their iris against their will [3, 19]. An ideal PAD technique should be able to detect all of these PAs along with any new or unknown PAs that may be developed in the future. Although a number of iris PAs have been described, current literature and publicly available datasets focus primarily on three main PAs — printed images, plastic eyes, and cosmetic contacts [28, 8, 16, 32, 33, 4, 6] — possibly due to their ease of construction and affordability.

PAD methods can be divided into two overarching categories: hardware-based and software-based. Hardware methods employ additional sensors or equipment, besides the iris device itself, in order to detect a PA. Software methods, on the other hand, use the image and other related information acquired by the iris device in order to detect a PA. Several hardware solutions have been proposed in the literature. Pacut and Czajka [3, 19] used the pupil's natural reaction to light stimulus to detect printed images, while Connell *et al*. [2] used structured light to distinguish between the 3D shape of live irides and cosmetic contacts. Both Lee *et al*. [15, 16] and Park and Kang [20] examined the differences between the reflected light of live and PA samples under multiple bands of near-infrared (NIR) light. Komogortsev *et al*. [14, 25] used eye tracking hardware to distinguish between the movements of a live eye and that of a printed image.

A number of software-based methods have also been proposed in the literature. Many researchers have investigated the use of local texture descriptors — such as LBP, LPQ, and BSIF — in conjunction with a Support Vector Machine or other classifiers [9, 31, 22, 11, 6, 5, 13]. Image quality features [8] and image frequency analysis [19] have also been used to develop iris PAD schemes. Menotti *et*

*Corresponding Author

[1]Note that early literature on this topic often used the phrase *spoof attack* in lieu of *presentation attack*.



(a) Print (b) Plastic (c) Cosmetic Contacts

Figure 1: Examples of artifacts used to launch iris presentation attacks (PAs). We show here (a) printed images, (b) plastic eyes, and (c) cosmetic contacts [12].

al. [18], He *et al*. [10], and Raghavendra *et al*. [23] trained convolutional neural networks (CNNs) on iris images and on patches from normalized iris images in order to develop iris PAD solutions. While most of these methods resulted in very high PA detection rates, they were primarily evaluated in intra-dataset scenarios where training and testing were based on the same types of PAs. Evaluation on sensors and PAs that were not used in the training set showed a dramatic decrease in detection accuracy [6, 31, 5, 23].

In this paper, we develop an iris PAD method that performs well in both intra-dataset and cross-dataset scenarios. It advances the state-of-the-art by considering the cross-dataset evaluation scenario that has received very little attention in the iris biometrics literature.

2. Proposed Method

In recent years, CNNs have achieved state-of-the-art performance on many computer vision tasks, including biometrics [29, 21]. Nonetheless, the literature on using CNNs for iris PAD is relatively sparse [18, 10, 23]. Further, no analysis has been conducted to determine the generalizability of these CNNs across different types of PAs and sensors. Here, we discuss the design of our iris PAD CNN in order to improve upon the existing work while also showing the cross-dataset capabilities of CNNs.

2.1. Data Preprocessing

Iris datasets used in biometrics research typically contain images that exhibit additional ocular details besides the iris, as seen on the left of Figure 2. However, in some PAs, such as cosmetic contacts, only the iris region will contain the PA information; the rest of the ocular region is unlikely to manifest any trace of the artifact. Therefore, we segment and localize the iris region to reduce confounding ocular information. The USIT segmentation tool [24] was used to facilitate segmentation whenever needed in this work. Since the iris size in an ocular image varies significantly within and across datasets, we resize all cropped iris images to

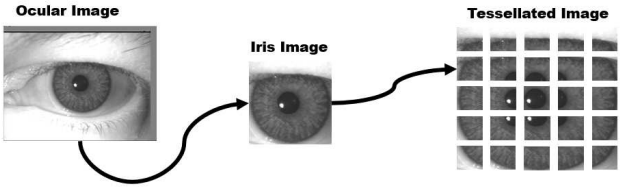


Figure 2: An overview of the data preprocessing used in this work. An ocular image (left) is cropped to the iris region and resized to 300×300 pixels (middle) before 25 overlapping patches of size 96×96 are extracted. This image containing a cosmetic contact lens is taken from the NDCC13 dataset [4].

300×300 pixels, thereby offering a consistently sized input to the learning algorithm. A size of 300×300 was chosen so that the vast majority of images considered in this work would be upsampled during resizing, as downsampling causes potentially important information to be lost. Finally, we tessellate the segmented and resized iris image into 25 overlapping patches of size 96×96. The primary reason for this tessellation is for data augmentation; many of the iris PAD datasets do not contain sufficient number of data samples to effectively train a deep network. Square patches are used to capture local texture information without having to make assumptions on the type of PA being presented. We tested three different patch sizes — 48×48, 96×96, and 128×128 — and found that 96×96 patches perform the best; see Section 4.3 for more details. The patches overlap and, therefore, if important textural information were to lie along the edge of one patch, they will be fully contained in the adjacent patch. The whole preprocessing pipeline is displayed in Figure 2. Finally, after obtaining a PA score for each of the 25 patches, we employ various fusion techniques, detailed in Section 2.3, to obtain a single PA score for the iris.

Novelty with respect to other CNN-based schemes:

Our preprocessing differs from that used for CNNs elsewhere [18, 10, 23]. Menotti *et al*. input the full ocular image into their CNN as they were only trying to identify paper prints [18]. This, however, would not be appropriate for identifying certain PAs, such as cosmetic contacts, which only affect the iris region. He *et al*. [10] and Raghavendra *et al*. [23] input patches of the normalized[2] iris image to their CNNs. In order to retain as much information in the data as possible, we choose to input *patches* of the *unnormalized* iris.

2.2. CNN Design Rationale

Our CNN design utilizes a set of eight convolutional layers, four max pooling layers, and a fully connected layer with a ReLU non-linearity function following each convolutional layer. The CNN takes as input a single iris patch, as detailed in Section 2.1, and outputs a PA score. A PA score is a number in the range $[0, 1]$ with 1 indicating a high confidence that the input contains a PA. A Euclidean loss function was chosen to train this network instead of the traditional softmax loss since the former allows for the direct calculation of confidence scores, rather than just generating class labels. Furthermore, our preliminary tests indicated that the Euclidean layer performed better than the softmax layer. The CNN architecture can be seen in Figure 3. The design of our CNN is inspired by the state-of-the-art VGG net [27]. However, to account for the small size of the input iris patch and the availability of limited training data, we used a shallow version of the VGG net.

Our CNN model also judiciously accounts for the number of iris and pupil pixels present in the input patch. When detecting iris presentation attacks, it is important to account for this since we focus primarily on attacks that can spoof or obfuscate the iris region. For instance, an adversary launching a printed iris attack may cut out the pupil region from the printed iris and place the print in front of their eyes in order to confound sensors that look for specular reflection in the pupil region [25]. Similarly, attacks based on cosmetic contacts may not obscure the pupil. Thus, pupil pixels seem less likely to supply useful information for discriminating PAs from live samples[3]. To accommodate this observation, we added a second channel to our CNN's input which we refer to as a *segmentation mask* (*segmask*). We define a segmask as a 2-dimensional matrix that is of the same size as the input iris patch. For every pixel location that corresponds to the iris region in the patch, the segmask contains a value of $+1$, and for every pixel corresponding to a the pupil region, the segmask contains a value of -1. For all other pixels in the patch (e.g. sclera, eyelids, etc.) the segmask contains a value of 0. By adding this segmask as a second input channel, the CNN can learn the importance of each region of the iris image automatically without introducing any additional human bias. An example segmask is shown in Figure 3.

The key aspects of our CNN model are summarized here: (1) The CNN takes iris *patches* as input rather than the full iris or ocular image thereby facilitating data augmentation during the training phase. (2) The input iris patches are taken from the *unnormalized* iris image rather than the normalized iris to avoid the downsampling that occurs during iris normalization. (3) A single CNN is trained on patches originating from all parts of the cropped iris image, and, as such, the CNN does not attempt to learn *location* artifacts but focuses on PA artifacts. (4) *Domain-specific* knowledge is incorporated by accounting for the number of iris and pupil pixels in an input patch through the inclusion of segmentation masks in the input and through defining patch-level fusion functions, as will be seen in Section 2.3.

2.3. Fusion Techniques

As described in Section 2.1, each ocular image is tessellated into 25 patches, and each one of these 25 patches produces its own score after passing through the CNN. However, a fusion method is needed in order to consolidate the 25 scores and render a decision. One possible fusion method is to take the *average score*, $s_{av} = \frac{1}{K}\sum_{i=1}^{K} s_i$, where s_{av} is the average score, s_i is the score of the i^{th} iris patch, and K is the total number of patches per image ($K = 25$ in this case). Note that $s_{av} \in [0, 1]$, where 0 indicates a live sample and 1 indicates a PA.

As mentioned in Section 2.2, it is likely that the percentage of iris and pupil pixels in a patch will affect the PA score. We designed two score fusion techniques based on this intuition, the iris-only ratio (io) score and the iris-pupil ratio (ip) score:

$$s_{io} = \left[\left(\frac{1}{\sum_{i=1}^{K} a_i} \right) \sum_{i=1}^{K} [s_i]_{[-1,+1]} * a_i \right]_{[0,1]}, \tag{1}$$

$$s_{ip} = \left[\left(\frac{1}{\sum_{i=1}^{K} \frac{a_i}{1+b_i}} \right) \sum_{i=1}^{K} [s_i]_{[-1,+1]} * \left(\frac{a_i}{1+b_i} \right) \right]_{[0,1]}, \tag{2}$$

where, s_{io} and s_{ip} are the iris-only ratio and iris-pupil ratio scores, respectively, s_i is the score of the i^{th} iris patch, a_i and b_i are the proportion of iris pixels and the ratio of pupil pixels in the i^{th} patch, respectively, K is the total number of patches per image, $[\cdot]_{[-1,+1]}$ is a function that converts scores to a $[-1, +1]$ range, and $[\cdot]_{[0,1]}$ is a function

[2]Here, normalization refers to the *unwrapping* of the iris wherein it is mapped from Cartesian coordinates to Pseudo-polar coordinates resulting in a fixed-size rectangular entity.

[3]The term "live" is used to indicate that the iris being presented is real and unmodified. In some literature, this is referred to as "bonafide".

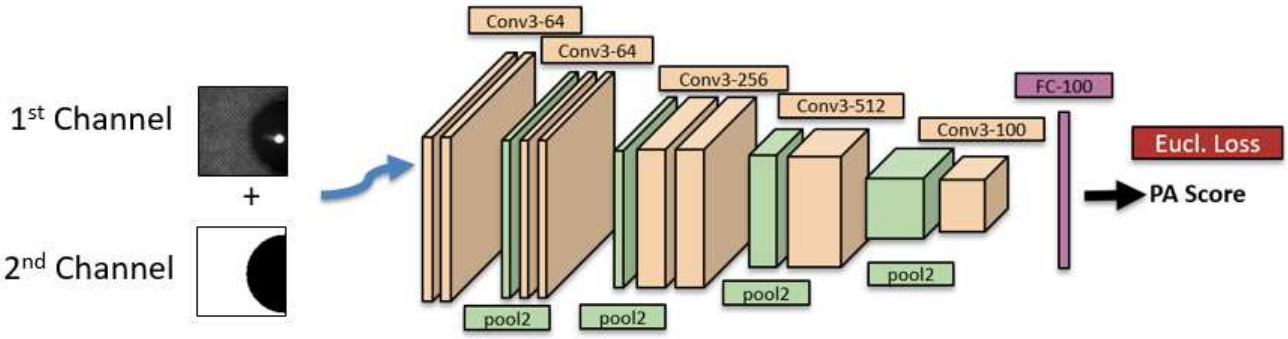


Figure 3: The CNN architecture used in this paper. Note that a ReLU layer follows each convolutional layer in this diagram. The first input channel represents a patch of the iris image while the second input channel contains the associated segmentation mask, as described in Section 2.2.

that converts scores to a $[0, 1]$ range. During this computation, we first convert the scores to a $[-1, +1]$ range so that patches normally having a score of 0 will now have a score of -1 and will be affected by the ratio of iris and pupil pixels. We divide by $1 + b_i$ rather than b_i itself to account for those cases where $b_i = 0$. This produces the iris-only (io) score, which is a weighted average giving higher priority to a patch's score if it contains a larger proportion of iris pixels, and the iris-pupil (ip) score, which is a weighted average giving higher priority to a patch's score if it contains more iris pixels and less priority if it contains more pupil pixels. These fusion techniques do not directly take into account the spatial location of the patch; rather they only consider the proportion of iris or pupil contained in a patch.

2.4. Evaluation Metrics

We report the True Detection Rate (TDR) and False Detection Rate (FDR), as defined below:

$$TDR = \frac{\text{\# Correctly classified PA images}}{\text{Total \# PA images}}, \quad (3)$$

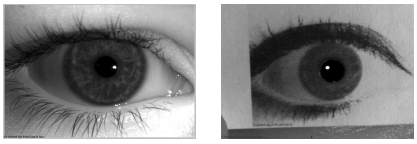
$$FDR = \frac{\text{\# Live images misclassified as PAs}}{\text{Total \# Live images}}. \quad (4)$$

3. Datasets

The proposed method was evaluated on three publicly available datasets: the LivDet-Iris 2015 Warsaw dataset, the CASIA-Iris-Fake dataset, and the BERC-Iris-Fake dataset. Since segmentation information was not provided with all these datasets, we used automatic segmentation software to locate the irides and pupils in the images. Whenever the software failed to locate the iris and pupil within an image, it was removed from the dataset.[4] Table 1 summarizes the number of images that were originally available in each of these datasets and the number that remained after excluding those images which were not segmented. When training the CNNs, the training partition of each dataset was first balanced to ensure an equal number of live and PA samples; this was accomplished by randomly removing an appropriate number of training samples from the larger class. The rest of this section gives further details on each dataset.

[4]We chose to discard these images, rather than manually segment them, in order to have a fully automated PAD system. Future work will handle

3.1. LivDet-Iris 2015 Warsaw Dataset



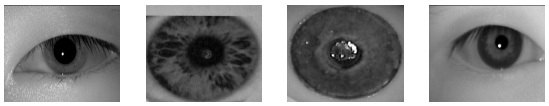
(a) Live (b) Print

Figure 4: Example images from the LivDet-Iris 2015 Warsaw dataset.

The LivDet-Iris 2015 dataset was developed for the 2015 Iris Liveness Detection Competition [33]. The Warsaw subset used in this work contains images of both live irides and printed PAs. Note that the authors of this dataset claim that it should contain 852 live images and 815 printed images for training [33]; however, due to an error during data acquisition, we only gained access to 603 live and 582 printed images from this set. Since the missing images were not acquired at the time of writing this paper, we trained only on a subset of the LivDet-Iris Warsaw 2015 dataset. Example images from this dataset can be seen in Figure 4.

Table 1: Summary of the datasets used in the paper along with the number of images in each dataset, the number of images on which automatic segmentation failed and the number of images remaining after those with failures were removed. Images are divided into live samples and the three addressed PAs: printed images, plastic eyeballs, and cosmetic contacts (CC).

		Training				Testing			
		Live	**Print**	**Plastic**	**CC**	**Live**	**Print**	**Plastic**	**CC**
LivDetW15	**Original**	603	582	-	-	2002	3890	-	-
	Seg Fails	0	0	-	-	30	2	-	-
	Remaining	603	582	-	-	1972	3888	-	-
CASIA-IF	**Original**	4800	512	320	592	1200	128	80	148
	Seg Fails	153	3	0	163	39	0	0	19
	Remaining	4647	509	320	429	1161	128	80	129
BERC-IF	**Original**	2258	1280	40	100	520	320	40	40
	Seg Fails	609	683	9	53	194	209	20	29
	Remaining	1649	597	31	47	326	111	20	11



(a) Live (b) Print (c) Plastic (d) Contact

Figure 5: Example images from the CASIA-Iris-Fake dataset.

(a) Live (b) Print (c) Plastic (d) Contact

Figure 6: Example images from the BERC-Iris-Fake dataset.

3.2. CASIA-Iris-Fake Dataset

The CASIA-Iris-Fake dataset [28] is another dataset that has been used for iris PAD evaluation. In contrast to the LivDet Warsaw set, which only contains the printed iris PA, the CASIA-Iris-Fake contains three of the most commonly discussed PAs in the literature: printed, plastic, and cosmetic contacts. Note that CASIA does not provide separate train and test partitions; rather, all the images are grouped together in a single partition. Therefore, we partitioned the dataset ourselves into training and testing subsets. Since subject information was not provided, we used visual inspection of images and corresponding file names to do our best in ensuring that subjects in the training and test sets were mutually exclusive. Also, the images in this dataset were already pre-cropped to the iris region. Example images from this dataset can be seen in Figure 5.

3.3. BERC-Iris-Fake Dataset

The BERC-Iris-Fake dataset [15, 16] also contains three of the most commonly used PAs in the literature — printed, plastic, and cosmetic contacts — making it useful for iris PAD evaluation. This dataset is unique for two reasons. First, rather than providing images captured at only a single NIR wavelength, the BERC dataset provides images at two separate wavelengths: 750nm and 850nm; we pool the images from these two spectral bands into a single set. Second, this dataset it has two separate types of print attacks: those created using an inkjet printer and those created using a laserjet printer; we pool these two sets of images into one set as well. This dataset also provides an additional challenge in that its images appear to be preprocessed. Visual inspection shows that all the printed and contact PAs have a black circle placed over the pupil to mask both the pupil and any specular reflections that may have otherwise been seen in the pupil region. Furthermore, a transparent black circle with a synthesized specular reflection has been placed over the pupil of about half of the plastic PA images. Finally, this dataset does not have separate training and testing subsets, so we partitioned the data ourselves using the same procedure as for the CASIA-Iris-Fake dataset. Example images from this dataset can be seen in Figure 6.

these discarded images in an automated fashion.

4. Results

We trained a CNN on each dataset separately[5] using the MatConvNet toolkit [30], experimenting with different batch sizes and learning rates to optimize performance. When a dataset contained multiple PAs (i.e., CASIA and

[5]This resulted in three CNNs, which we refer to as LivDet-CNN, CASIA-CNN and BERC-CNN in this paper. Note that each CNN was trained using only the training partition of the corresponding dataset.

BERC), we trained a single CNN on all PA types to increase the model's generalizability across PA types. We also balanced the number of live and PA samples during training. During testing, however, we evaluated the performance of each PA type separately to obtain more insight into the performance of our CNN on each type of PA.[6]

We report TDR values at a stringent FDR value of 0.2%, using the iris-pupil ratio fusion technique as described in Section 2.3. We chose this fusion technique as it produced better results than both the average score and the iris-only ratio score on the LivDet and BERC datasets. On the CASIA dataset, the average score performed the best and, therefore, this score is reported on intra-dataset tests involving CASIA.

In the case of the BERC-CNN, we found that the automatic segmentation results on the training set were poor, not being able to localize the iris and pupil as proficiently compared to the other two datasets. This caused the CNN trained with segmentation masks to perform worse than the same CNN trained without segmentation masks. The results on BERC are, therefore, from the CNN without using the segmask.

When a CNN is tested on the same dataset that it was trained on, we call this an *intra-dataset* evaluation; when the same CNN is evaluated on a dataset that is different from what it was trained on, we call this a *cross-dataset* evaluation. In all cases, subjects in the training and test sets are mutually exclusive.

4.1. Intra-Dataset Results

Our intra-dataset results[7] are shown in Table 2. The authors of the Federico model [33] had access to the complete LivDet-Iris 2015 training set, meaning they were able to train on a larger dataset than we could. Nonetheless, our model is able to achieve comparable results with the current state-of-the-art, showing the efficacy of our model. Furthermore, the authors of the BERC-H model [16] used a hardware-based method to perfectly classify the BERC-Iris-Fake dataset, relying on a sensor to capture images at both 750nm and 850nm in order for their system to classify it as live or spoof. We were able to replicate their accuracy without taking into account the multiple spectra, using a software-based solution. Our model, however, was unable to reproduce the state-of-the-art for the CASIA-Iris-Fake dataset at the stringent FDR of 0.2%, particularly failing at identifying cosmetic contacts. Upon analyzing the CASIA dataset, we found that it exhibits a higher degree of variability in the PA samples than the LivDet and BERC datasets. In particular, it consisted of a large number of different cosmetic contact styles. We speculate that our CNN model may have been too shallow to efficiently capture this variability, causing the decreased performance on CASIA.

Table 2: The results of the proposed approach on each dataset under intra-dataset conditions. We evaluated against each PA type separately to get a better understanding of how our CNNs work on each PA. The TDR@FDR$= 0.2\%$ is reported for print, plastic, and cosmetic contact (CC) PAs.

	Print	**Plastic**	**CC**
LivDetW15*	99.87%	-	-
CASIA-IF*	89.84%	95.00%	24.81%
BERC-IF*	100%	100%	100%

4.2. Cross-Dataset Results

In the literature, very little has been reported on the generalizability of iris PAD algorithms across PAs, sensors, or datasets. A few authors have attempted *cross-sensor* testing of their PAD algorithms [5, 23, 31]. These authors evaluated their models on datasets that used multiple sensor brands. They trained their model on data from one sensor and evaluated on data from the other sensors. Under these scenarios, the TDR was often much lower than those seen in intra-dataset conditions, and the FDR was much higher. This highlights the difficulties in cross-sensor testing scenarios.

However, an even more difficult testing condition than this is to perform *cross-dataset* testing. During cross-sensor testing, one must primarily account for variations in cameras, but during cross-dataset testing, one must account for variations in the sensors, data acquisition environment, subject population, and PA generation procedures. This is referred to as dataset biases [7, 1]. This makes cross-dataset testing a very difficult problem, and to the best of our knowledge, cross-dataset evaluation has not been conducted so far for iris PAD methods. Nonetheless, a real-world scenario demands that iris PAD algorithms be able to operate under the kind of variations seen in cross-dataset testing. To perform cross-dataset evaluation, we took our best performing CNN from the intra-dataset testing scenario, viz., the BERC-CNN, and tested it against the other two datasets. The results can be seen in Table 3. This CNN achieved high TDRs on both the LivDet-Iris Warsaw 2015 dataset and the printed PAs of the CASIA-Iris-Fake dataset, returning TDRs of 95.11% and 100%, respectively, at an FDR of 0.2%. The model did not perform as well on the plastic or contact PAs of the CASIA dataset, achieving TDRs of 43.75% and 9.30% respectively. This suggests that *cross-PA* testing represents a very challenging scenario.

[6] Testing was conducted only on the test partition of the corresponding dataset.

[7] Note that we report results from training and testing only on those images for which automatic segmentation was successful. We include an asterisk next to the dataset name to indicate this.

4.3. Patch Size Analysis

In order to study the impact of patch size, we repeated the above experiments, which use a patch size of 96×96, with patch sizes of 48×48 and 128×128. We trained these CNNs with batch normalization to improve convergence.

The results of the tests on the different patch sizes are shown in Table 4. From this table, we can see that the patch size of 96×96 provides the best results. The performance increases when expanding the patch size from 48×48 to 96×96, possibly because the CNN has more information in a patch to generate a PA score. However, when increasing the patch size from 96×96 to 128×128, the number of weights needed to train the CNN increases and the amount of data we have available to train the network is likely too small, thereby decreasing the accuracy on 128×128 patches.

4.4. Feature Map Analysis

Table 3: The results of the proposed approach under cross-dataset conditions when having trained on BERC-Iris-Fake* and tested on the remaining two datasets. The TDR@FDR= 0.2% is reported for print, plastic and cosmetic contact (CC) PAs.

Test Set	Print	Plastic	CC
LivDetW15*	95.11%	-	-
CASIA-IF*	100%	43.75%	9.30%

Table 4: Impact of varying patch sizes on iris PAD. The TDR@FDR=0.2% is reported for CNNs trained on the BERC-Iris-Fake* dataset.

Test Set	PA Type	48x48	96x96	128x128
BERC-IF*	**Print**	100%	100%	100%
BERC-IF*	**Plastic**	100%	100%	50.00%
BERC-IF*	**CC**	100%	100%	90.91%
LivDetW15*	**Print**	68.03%	95.11%	75.03%
CASIA-IF*	**Print**	49.22%	100%	0.00%

To gain a better understanding of what our CNN models learned, we analyzed the intermediate representations of image patches as they passed through the CNN. We looked at our best performing model, the CNN trained on the BERC dataset, and viewed the outputs (feature maps) of the 1st, 3rd, 5th, and 7th convolutional layers. Exemplar feature maps for a PA sample can be found in Figure 7, while those for a live sample can be seen in Figure 8. An analysis of these images reveals that the first few layers of the CNN work as edge detectors, emphasizing edges formed by both the iris texture and by the low resolution of the PA.

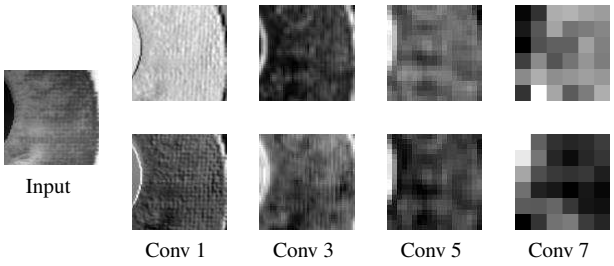


Figure 7: An image patch from a BERC cosmetic contact PA sample, and the feature maps that are generated as it passes through the 1st, 3rd, 5th and 7th convolutional layers of the CNN trained on the BERC dataset.

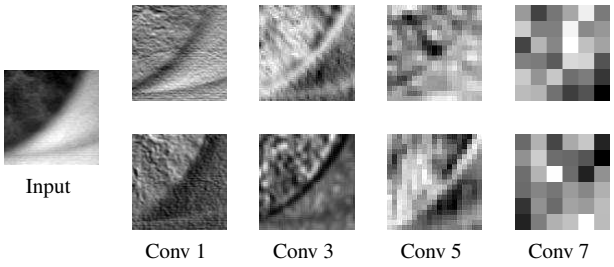
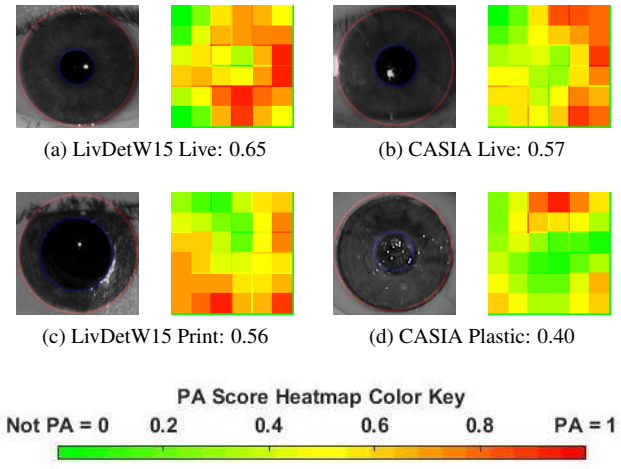


Figure 8: An image patch of a live iris image from the BERC dataset, and the feature maps that are generated as it passes through the 1st, 3rd, 5th and 7th convolutional layers of the CNN trained on the BERC dataset.

In particular, the outputs of the 1st and 3rd convolutional layers for the PA sample reveal a fine-grained grid pattern due to the pixelation in the cosmetic contact lens. These pixelations get smoothened out later in the CNN such that by the 7th convolutional layer, the output feature maps exhibit only small variations in intensity. For the live samples, however, a strong grid pattern is not seen in the first few feature maps. This leads the CNN to exhibit large variations in image intensity in the feature maps corresponding to the 7th layer, allowing the live samples to be discriminated from PAs. This analysis suggests that our model should do well at identifying PAs whose iris details are of lower resolution. Conversely, when a PA sample has a higher resolution and the inherent pixelation is not evident in the resulting image, our model is not as likely to do well in PA detection.

4.5. Failure Case Analysis

To better understand our CNN trained models, we visually analyzed the misclassified images by the BERC CNN on the cross-dataset experiments. We generated heatmaps for each image, where the intensity of a pixel in this heatmap corresponded to the average PA score of all patches embodying that pixel. These heatmaps allowed us to better



(a) LivDetW15 Live: 0.65 (b) CASIA Live: 0.57

(c) LivDetW15 Print: 0.56 (d) CASIA Plastic: 0.40

Figure 9: Images misclassified by the CNN trained on BERC. The corresponding heatmaps, ground truth classification labels, and PA scores are also shown. Thresholds of 0.64 for testing on LivDet and 0.57 for testing on CASIA were used to obtain the desired FDR of 0.2%.

visualize the differences in PA detection accuracy across an image. Example heatmaps can be seen in Figure 9. A few observations can be made. First, we noticed that a large number of errors in both live and PA images occurred when there was a significant amount of glare. Since glare obscures the underlying object's texture, this artifact makes PA classification much more difficult. We also noticed that in some of the misclassified live images, such as the one in Figure 9a, the texture of the iris was rather smooth, not containing a lot of discriminative texture. The same can be seen in some misclassified spoof images, like in Figure 9c, where the PA sample has high contrast, making it difficult to discern any texture. This is consistent with our feature map analysis, suggesting that when neither the irregular texture of a live iris nor the fine-grained grid pattern of PAs can be identified by the CNN's edge-detecting filters, the CNN is unable to reliably classify the image.

5. Discussion

Upon analyzing the cross-dataset results, we see that the proposed CNN was able to perform exceptionally well on the LivDet-Iris Warsaw 2015 dataset and the printed PA subset of the CASIA-Iris-Fake dataset. The CNN was able to harness the power of deep learning to generate feature extractors that generalize well across datasets. In addition, because the BERC-Iris-Fake dataset contains images at two different light spectra, we believe that the CNN was able to extract more variations than that which naturally occurs when training on only a single light spectrum. However, the CNN did not perform well on plastic or contact PAs from the CASIA dataset at an FDR of 0.2%. For plastic PAs, this is likely because the BERC dataset only contains 31 plastic images for training, so it was unable to learn a model strong enough to identify this PA in another dataset, especially since the plastic eye images in the CASIA dataset look much different than those used in BERC (compare Figure 5 and Figure 6). For the cosmetic contact PA [5], we have already noted that the variations in the contacts in the CASIA dataset are much higher than those in the BERC dataset, accounting for a much lower TDR in both intra-dataset and cross-dataset testing. Nonetheless, our proposed model was able to establish state-of-the-art results in the domain of cross-dataset iris PAD testing.

Acknowledgment

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] S. Banerjee and A. Ross. From Image to Sensor: Comparative Evaluation of Multiple PRNU Estimation Schemes for Identifying Sensors from NIR Iris Images. In *Fifth International Workshop on Biometrics and Forensics*, 2017. 6

[2] J. Connell, N. Ratha, J. Gentile, and R. Bolle. Fake Iris Detection Using Structured Light. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8692–8696, 2013. 1

[3] A. Czajka. Iris Liveness Detection by Modeling Dynamic Pupil Features. In M. J. Burge and K. W. Bowyer, editors, *Handbook of Iris Recognition*, volume 1542, chapter 19, pages 439–467. Springer-Verlag, London, 2013. 1

[4] J. Doyle and K. Bowyer. Technical Report: Notre Dame Image Database for Contact Lens Detection in Iris Recognition-2013, 2014. 1, 2

[5] J. S. Doyle and K. W. Bowyer. Robust Detection of Textured Contact Lenses in Iris Recognition Using BSIF. *IEEE Access*, 3:1672–1683, 2015. 1, 2, 6, 8

[6] J. S. Doyle, K. W. Bowyer, and P. J. Flynn. Variation in Accuracy of Textured Contact Lens Detection Based on Sensor and Lens Pattern. *IEEE 6th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013. 1, 2

[7] S. El-Naggar and A. Ross. Which Dataset is this Iris Image From? In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Nov 2015. 6

[8] J. Galbally, S. Marcel, and J. Fierrez. Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition. *IEEE Transactions on Image Processing*, 23(2):710–724, 2014. 1

[9] 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. 1

[10] L. He, H. Li, F. Liu, N. Liu, Z. Sun, and Z. He. Multi-patch Convolution Neural Network for Iris Liveness Detection. *IEEE 8th International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, 2016. 2, 3

[11] Z. He, Z. Sun, T. Tan, and Z. Wei. Efficient Iris Spoof Detection via Boosted Local Binary Patterns. *International Conference on Biometrics*, 1(c):1087–1097, 2009. 1

[12] Hoover Vision Center. Halloween Hazard: The Dangers of Cosmetic Contact Lenses, http://hoovervisioncenter.com/2015/10/21/halloween-hazard-the-dangers-of-cosmetic-contact-lenses/, visited: 2018-01-03. 2

[13] N. Kohli, D. Yadav, M. Vatsa, R. Singh, and A. Noore. Detecting Medley of Iris Spoofing Attacks using DESIST. In *IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6, 2016. 1

[14] O. V. Komogortsev, A. Karpov, and C. D. Holland. Attack of Mechanical Replicas: Liveness Detection with Eye Movements. *IEEE Transactions on Information Forensics and Security*, 10(4):716–725, 2015. 1

[15] S. J. Lee, K. R. Park, and J. Kim. Robust Fake Iris Detection Based on Variation of the Reflectance Ratio between the Iris and the Sclera. *BCC Biometrics Symposium*, 2006. 1, 5

[16] S. J. Lee, K. R. Park, Y. J. Lee, K. Bae, and J. Kim. Multifeature-based Fake Iris Detection Method. *Optical Engineering*, 46(12):127204, 2007. 1, 5, 6

[17] S. Marcel, M. S. Nixon, and S. Z. Li. *Handbook of Biometric Anti-Spoofing*. 2014. 1

[18] 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, 3

[19] A. Pacut and A. Czajka. Aliveness Detection for Iris Biometrics. In *40th Annual IEEE International Carnahan Conferences on Security Technology (ICCST)*, pages 122–129. IEEE, 2006. 1

[20] J. H. Park and M. G. Kang. Iris Recognition Against Counterfeit Attack Using Gradient Based Fusion of Multi-spectral Images. *Engineering*, pages 150–156, 2005. 1

[21] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In *British Machine Vision Conference BMVC*, volume 1, page 6, 2015. 2

[22] 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, 2015. 1

[23] R. Raghavendra, K. B. Raja, and C. Busch. ContlensNet: Robust Iris Contact Lens Detection Using Deep Convolutional Neural Networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1160–1167. IEEE, 2017. 1, 2, 3, 6

[24] C. Rathgeb, A. Uhl, P. Wild, and H. Hofbauer. Design Decisions for an Iris Recognition SDK. In K. Bowyer and M. J. Burge, editors, *Handbook of Iris Recognition*, Advances in Computer Vision and Pattern Recognition. Springer, second edition edition, 2016. 2

[25] I. Rigas and O. V. Komogortsev. Eye Movement-driven Defense Against Iris Print-attacks. *Pattern Recognition Letters*, 68(July):316–326, 2015. 1, 3

[26] A. Ross. Iris Recognition: The Path Forward. *IEEE Computer*, 43(2):30–35, Feb 2010. 1

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3

[28] 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, 2014. 1, 5

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the Gap to Human-level Performance in Face Verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014. 2

[30] A. Vedaldi and K. Lenc. MatConvNet – Convolutional Neural Networks for MATLAB. In *ACM Int. Conf. on Multimedia*, 2015. 5

[31] 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, 2014. 1, 2, 6

[32] D. Yambay, J. S. Doyle, K. W. Bowyer, A. Czajka, and S. Schuckers. LivDet-Iris 2013-Iris Liveness Detection Competition 2013. pages 1–8. IEEE, 2014. 1

[33] D. Yambay, B. Walczak, S. Schuckers, and A. Czajka. LivDet-Iris 2015 - Iris Liveness Detection Competition 2015. In *IEEE International Conference on Identity, Security, and Behavior Analysis (ISBA)*. IEEE, 2017. 1, 4, 6