Neural Computing and Applications (2021) 33:5609–5629
https://doi.org/10.1007/s00521-020-05342-3

ORIGINAL ARTICLE

Iris presentation attack detection based on best-*k* feature selection from YOLO inspired RoI

Meenakshi Choudhary[1] · Vivek Tiwari[1] · Venkanna Uduthalapally[1]

Received: 16 November 2019 / Accepted: 3 September 2020 / Published online: 22 September 2020

Abstract
Obfuscating an iris recognition system through forged iris samples has been a major security threat in iris-based authentication. Therefore, a detection mechanism is essential that may explicitly discriminate between the live iris and forged (attack) patterns. The majority of existing methods analyze the eye image as a whole to find discriminatory features for fake and real iris. However, many attacks do not alter the entire eye image, instead merely the iris region is affected. It infers that the iris embodies the region of interest (RoI) for an exhaustive search towards identifying forged iris patterns. This paper introduces a novel framework that locates RoI using the YOLO approach and performs selective image enhancement to enrich the core textural details. The YOLO approach tightly bounds the iris region without any pattern loss, where the textural analysis through local and global descriptors is expected to be efficacious. Afterward, various handcrafted and CNN based methods are employed to extract the discriminative textural features from the RoI. Later, the best-*k* features are identified through the Friedman test as the optimal feature set and combined using score-level fusion. Further, the proposed approach is assessed on six different iris databases using predefined intra-dataset, cross-dataset, and combined-dataset validation protocols. The experimental outcomes exhibit that the proposed method results in significant error reduction with the state of the arts.

Keywords DarkNet-19 · Feature selection · Image enhancement · Iris presentation attack detection · RoI localization · Score-level fusion

1 Introduction

Iris recognition (IR) has achieved vigorous research interest due to its peerless individualities such as the rich morphological structure, certain distinctiveness for individuals (even twins), and constancy in micro-features regardless of the growing age [1]. Nevertheless, the IR systems are susceptible to presentation attacks that attempt to emasculate the application security. These attacks represent the forged or deliberately designed iris patterns in front of the iris camera/sensor to obstruct the functioning of the IR system [2]. These may be used to register contrived irises, purposely obscure a party's trait, or even forge the iris pattern of another person [3]. There are several ways to reproduce the iris patterns, such as using textured contact lenses, printed iris images, artificial eyeballs, and playing iris images/videos on the LCD, and drug-prompted iris employment [2, 3] as depicted in Fig. 1. As the IR systems are progressively installed in precarious applications, e.g., border control, airport security, etc., there is an urge for some security means to recognize the presentation attacks. With this motivation, various presentation attack detection (PAD) mechanisms are introduced in the literature [2].

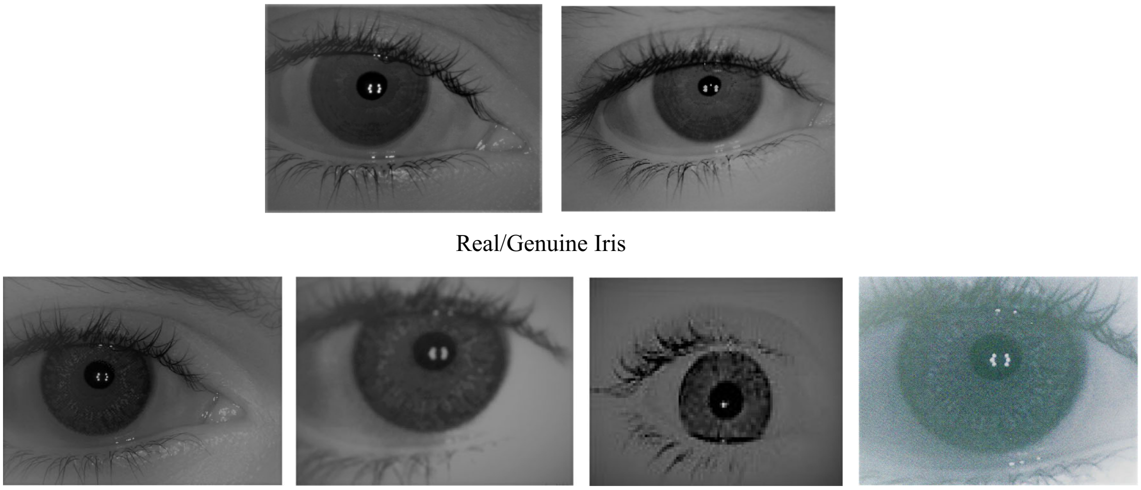
The current iris PAD approaches are categorized as either sensor-based or image-based. Sensor-based approaches generally incorporate additional hardware to acquire visual or physical patterns of the eye [4, 5]. Whereas, image-based methods analyze the micro-structures existing within the iris image through handcrafted methods and a

✉ Vivek Tiwari
vivek@iiitnr.edu.in

Meenakshi Choudhary
meenakshi@iiitnr.edu.in

Venkanna Uduthalapally
venkannau@iiitnr.edu.in

[1] DSPM IIIT Naya Raipur, Naya Raipur, CG 493661, India


Real/Genuine Iris

Textured Lens

Print + capture

Synthetic iris

Print + scan

Fig. 1 Depiction of various iris presentation attack samples (bottom row) corresponding to live iris (top row). The iris liveness detection methods are supposed to discriminate between these two categories

classifier [6]. Indeed, an image-based method takes an ocular image captured by the iris sensor, slices of the iris region, extracts local and/or global features, and categorizes it as "live" or "attack" through a classifier. There are several feature descriptors existing in the literature, such as local binary patterns (LBP) [7], binarized statistical image features (BSIF) [8], and scale-invariant descriptors (SID) [9], to constitute pixel-level features. The recent research is extensively utilizing the convolution neural networks (CNNs) for self-feature learning to realize PAD [10, 11]. The uncertainty of the micro-structures for live iris and attack samples results in various discrete patterns corresponding to the same class. Thus, a purposely developed handcrafted feature may be incapable of capturing all possible patterns [12]. Besides, several methods usually include iris segmentation as a crucial stage to perform local feature extraction from the segmented region [3]. Iris segmentation locates the inner and outer iris edges in the image. Nevertheless, it endures some problems since the structure of the iris is not essentially circular; in fact, it has no fixed shape. Therefore, detecting the iris edges without any pattern loss is extremely challenging. In this view, a PAD with such intrinsic segmentation techniques is not robust [12].

The Iris liveness detection competition began in 2013 to examine the evolving PAD algorithms, and to unveil the progress status of the iris PAD. The recent edition occurred in 2017 [10], which uncovered some interesting open problems, e.g., cross-sensor and cross-dataset systems (also known as cross-domain) in the context of iris PAD. With the aspect of improving the cross-domain iris PAD, this paper introduces an image-based PAD scheme as depicted in Fig. 2. It begins with the region of interest (RoI) localization, which is carried out by a preeminent CNN framework, i.e., DarkNet-19 [13] that was initially designed for generic object detection. This model predicts the spatial dimensions of the rectangular box that tightly bounds the RoI. The RoI is then cropped from the image based on the rectangle box, and then, we use OpenCV in python to detect the rectangle in the image and crop it. In the next step, the selective image enhancement is performed over the RoI of given iris images to remove blurriness and to magnify the pixel intensity [14, 15]. Further, the enhanced RoI is fed to various handcrafted and data-driven algorithms to extract key features and to produce corresponding feature-vectors. Further, an optimal feature set is obtained through the Friedman test based feature selection approach and is fused using score-level fusion for final attack prediction.

1.1 The motivation behind the proposed approach

This work is based on some important observations related to iris PAD. Most of the iris presentation attacks primarily alter the iris region rather than the entire eye image [2]. Moreover, the amendment caused by such attacks is also evident in the iris region. Based on such observations, it is concluded that analyzing the entire eye image for feature extraction is not desirable. Instead, it is beneficial to identify the region of interest (RoI) within the eye image, where best discriminative features exist corresponding to all possible presentation attacks [11]. The majority of existing iris segmentation approaches [16, 17] follow handcrafted procedures to perceive iris inner and outer boundary pixels. However, such procedures need a set of

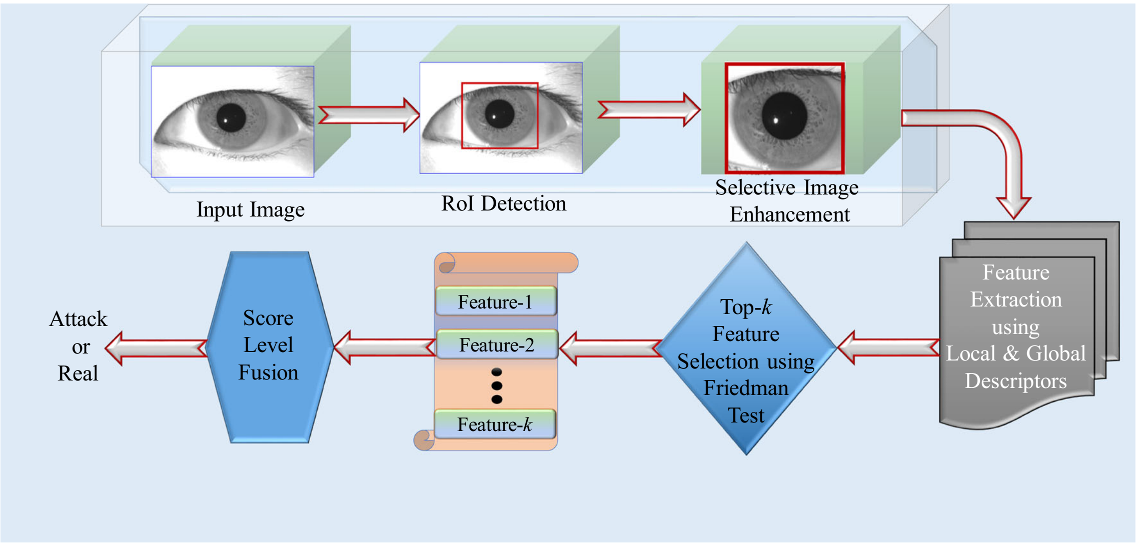

Input Image
RoI Detection
Selective Image Enhancement
Feature Extraction using Local & Global Descriptors
Top-k Feature Selection using Friedman Test
Feature-1
Feature-2
Feature-k
Score Level Fusion
Attack or Real

Fig. 2 Flowchart of the proposed approach. The region of interest (RoI) is identified using YOLO framework, then image enhancement is carried out on the selected region. Next, various local and global feature descriptors are employed to extract features from enhanced RoI. Further, top-*k* features are selected through Freidman test, and are combined using score-level fusion for attack prediction

empirically defined parameters and, thus, are less generalizable to varying image categories. Besides, the circular Hough transforms, and integro-differential operator for detecting iris and pupil boundaries work well with high-quality images but less robust to blur and noisy images [14]. In this vein, some approaches [12] apply filters on raw iris images for feature extraction, whereas others [3, 6] down-sample the images according to their favorable input size before feature extraction. However, both cases seem to be inadequate because feature extraction from raw images leads to a huge amount of computation and perhaps redundant features construction. A previous study [14] demonstrated a comparative study between RoI images and raw eye images using accuracy as the performance metrics, where accuracy on RoI images is substantially larger than raw images. They gave the reasoning that heavy eyelids and eyelashes may occlude the operative iris regions for feature extraction and may cause intra-class variations. Besides, image down-sampling in the required input size (e.g., VGG-16 requires the input size as 224 × 224) results in significant spatial feature loss [15]. Therefore, RoI detection and segmentation from the given eye image is an adequate choice for constituting better features. The choice of YOLO framework is inspired by the speed hike achieved at the test time as it analyses the given image only once, unlike other object localization models such as region proposal CNN and faster RCNN with repetitive image processing [18]. It also exhibits better generalizability and less error (less than half of the background errors compared to fast and faster RCNN) [19]. The speed is primarily important as the iris localization is an integral step of the test time procedure of the proposed approach.

The comprehensive study of computer vision and image classification suggests that multiple feature fusion substantially enhances the classification performances compared to the sole feature [20–22]. A similar concept is being followed in iris related applications, where handcrafted and data-driven features are combined to construct an enhanced feature set [12, 20, 22]. However, the entire feature set may encompass redundant or less important features, which should be removed to obtain an optimal feature set [23].

1.2 Contribution

YOLO- and CNN-based approaches have already been used for RoI detection and feature extraction for iris PAD. Even though, to the best of our knowledge, there is no previous work that focuses on optimal feature selection out of multiple handcrafted and data-driven features and deploying it in the cross-domain environment. The integration of handcrafted and data-driven features is aimed to exploit their respective benefits and to build an iris PAD algorithm with appropriate generalizability to various attack categories. The use of multiple algorithms provides key features extracted with diverse views to the data since each method examines the features with a distinct perspective. In specific, the use of a deep CNN model in RoI detection provides the flexibility to adapt to varying image qualities without extra parameter adjustments. Additionally, we have presented a novel insight into the preeminent Friedman test, where it may be used for optimal feature selection by examining each feature-vector with the corresponding output. The score-level fusion of optimal features yields a fair contribution of each feature in the attack prediction. The major contribution and novelties of this paper can be summarized as follows:

- A novel approach that employs the YOLO model for iris region localization since it is speedy and accurate in predicting RoI at the test time.
- An algorithm that employs multiple handcrafted and CNN-based methods for feature extraction in order to perceive key features with multiple perspectives.
- A new feature selection mechanism based on the Friedman test that examines each feature-vector with output labels on distinct databases to enhance the robustness of the optimal feature set.
- A comprehensive cross-domain assessment of the proposed PAD approach on datasets currently used to evaluate the state of the arts in the field of iris PAD.
- The proposed novel PAD approach outpaces the winner of LivDet-Iris-2017 (it is the most recently conducted iris liveness detection competition).

The leftover segment of this paper is structured as; Section 2 explores the literature regarding progress in iris PAD together with the current issues. Section 3 thoroughly describes the proposed scheme and various phases involved in processing the iris images. Section 4 describes the underlying datasets and validation protocols included for the proposed method assessment, along with the experimental outcomes and discussion. Finally, Section 5 concludes the entire work.

2 Related work summary

Since the last two decades, the vulnerability of an IR system to be obscured through presentation attacks has attained a sufficient interest of researchers. The presentation attack detection can be carried out at the sensor-level, pixel-level, or algorithm-level. At the sensor-level, specific designs of iris cameras/sensors can simplify live/fake iris detection. Lee et al. [5] addressed PAD through inspecting the specular blotches of collimated infrared light emitting diode (IR-LED). However, it is incapable of identifying contact lenses, as the visibility of iris texture worsens upon wearing it. Further, authors [24] incorporated algorithms based on pupil dynamics to perceive forged iris, and it failed to identify textured lenses and artificial irises. Sensor-level schemes may extensively acquire the ocular properties of the legitimate iris pattern. However, the generalization capability of the sensor-level PAD schemes is limited as they require the specific design of sensors and depend upon the special hardware functionalities.

Conversely, the pixel-level PAD schemes do not demand special iris sensors exploiting the optical features of iris to classify live/attack samples. Therefore, the techniques utilizing local descriptors to scrutinize iris micro-structures are extremely inspiring [25, 26]. In this context, Daugman [27] introduced the real-time PAD system, where extra peaks in the Fourier amplitude spectrum may be recognized via 2-D Fourier transforms for the cosmetic lens, which does not occur in the real iris's spectrum. He et al. [7] suggested utilizing local binary patterns (LBP) for contact lens-based PAD, where LBPs are undermined from six related iris subregions. Yet, the AdaBoost algorithm identifies the principal LBP feature. Several other feature descriptors, such as co-occurrence of adjacent LBP (CoA LBP) [28], DAISY [29], and HOG [30], have also been proposed in the literature. In addition, authors in [25] demonstrated an in-depth investigation of spoofing attacks on IR systems and employed multiscale BSIF for feature extraction from iris images. They primarily focused on printed iris images and iris video images captured from LCD. Furthermore, authors in [31] proposed a novel scheme to collect features from the regions of pupil and sclera. Here, LBPs are autonomously extracted from several regions of normalized iris images and then concatenated to discriminate between "attack" and "live" samples. Authors in [32] jointly utilized frequency analysis and extra quality features for printed iris and cosmetic lens detection. Moreover, Sharifi et al. [33] perceive cosmetics on iris and face images by exploiting the combination of micro-texton material and color spaces to discover edges, spots, curves, etc., magnificently. Additionally, the change in variability scores conveyed by fake and real texture is exploited as a distinctive factor. It is noticed that the aforesaid schemes include handcrafted feature extraction to produce iris codes and observes the deviation in attack samples from original counterparts. However, a similar effect could be attained by learning the attack patterns within the raw eye image. Precisely, matching the features or score distribution of fake irises with legitimate correspondents is not pretty adequate. Besides, in contrast to handcrafted features, self-feature learning could be adapted to identify attack patterns. Such approaches are described below.

An adequate framework adhering to the self-feature learning (data-driven) approach is the convolutional neural network (CNN). Menotti et al. [34] designed a three-layer CNN, namely SpoofNet as a liveness detection model for fingerprints, iris, and face. The model is capable enough to extract iris features and to undermine semantics and visual features from raw iris images. Next, an analogous architecture was used in [35] to discriminate among normal, soft, and textured lenses. However, such models conveyed diminished accuracy due to the shallow architecture. Further, in LivDet 2017 challenge [10], a seven-layer inception-based CNN architecture was participated to distinguish between live (real) and fake irises. Another multi-patch CNN model was proposed by He et al. [36], which is trained on 28 subsequent patches of real and attack samples. The respective outputs of all patches are gathered individually to feed the decision layer to classify between live and fake irises. However, the computational expenses are increased since a sole training stage entails 28 CNN operations. In addition, Choudhary et al. [6] exploited the DenseNet121 model with some customizations, for feature extraction and the SVM classifier for classification between iris contact lenses. However, these models map the given lens category images to the respective class, instead of considering the entire dataset. Besides, Chen et al. [11] introduced a multi-task CNN-based framework that concurrently detects iris region and presentation attacks in terms of probability. Notice that all these methods incorporate single feature extractor, i.e., the classification is carried out based on a single feature vector. However, as the presentation attacks may amend the real iris in several aspects, analyzing them with sole angle does not yield impressive results. Therefore, authors in [21] projected a premise that a pool of noble features results in an ominously enhanced discrimination. They anticipated a feature selection and fusion network using six different local features, i.e., LBP, HoG, CoA LBP, BSIF, SID, and DAISY, together with an eight-layer VGG model. However, the authors focused merely on textured lens-based PAD, while other categories of presentation attacks are left unexplored. Besides, Yadav et al. [22] delineated the feature-level fusion of VGG-8 and Haralick features for multiple iris presentation attack detection. Similarly, Kohli et al. [37] also fused the Zernike moment-based features with LBP with variance (LBPV) to handle the medley of iris presentation attacks. Furthermore, a recent work [38] suggested coalescing features from three distinct local and global regions within the given eye image through feature-level and score-level fusion. Table 1 comprehensively outlines the literature focusing on iris PAD in terms of the underlying feature extraction mechanisms.

3 Proposed approach: YOLO with statistical methods

This subsection describes each of the three subsequent phases of the proposed approach to improve iris liveness detection. Begin with region of interest (RoI) localization through bounding box regression; it demonstrates the architecture and functionality of the YOLO framework [13] used. Further, seven distinct feature extractors used to constitute features from enhanced RoI, and the Friedman test [39] used to accomplish best-*k* feature selection from resultant features are described in detail. Furthermore, the selected features are combined using score-level fusion to make a final attack prediction.

3.1 CNN framework for RoI detection

The RoI detection procedure is inspired by an earlier work [11] with a slight modification that it focuses on iris localization instead of liveness detection. The framework (model) deployed to detect RoI is depicted in Fig. 3, which adheres to a precise version of the CNN network, i.e., DarkNet-19 [13]. This framework contains nine convolutional blocks, where the first six are followed by max pooling layers. Each convolution block represents a combination of three subsequent operations, i.e, convolution, batch normalization (BN), and rectified linear unit (ReLU). Here, each convolution layer is implemented with a filter size of 3 × 3, excluding the last, where 1 × 1 filters are used. Notice that a fully connected layer is not included here so that the model automatically adapts to accept the varying sized input. The topmost layer uses a softmax function, which is responsible for predicting the coordinates of the bounding box representing the iris region. Table 2 shows the entire network architecture, along with several parameters. Here, the input image is resized with 416 × 416 × 3, before feeding to the model. Further, the spatial dimensions are diminished by a factor of 32 after performing a chain of convolution and pooling functions, and the output dimensions become 13 × 13 × 25. Indeed, the output feature map contains the number of channels as *(#class* + *#coords* + 1*)*#ancors*, where *#class* denotes the number of output classes, *#coords* represents the coordinates of the bounding box to predict (i.e., *x,y* coordinates depict the center of the bounding box, along with the height and width of the iris region), and *#ancors* is the number of predefined anchors (or the number of the bounding box to examine) to obtain the best bounding box. In the experiment, it is set to 5. Notice that, since merely single object detection and/or localization is carried out in the proposed setup, #class is set to 0, and the framework is utilized as a regressor instead of a classifier.

The DarkNet-19 model was originally trained with the ImageNet dataset with 1000 distinct output categories. However, this model is designed to act as a classifier. Therefore, it is retrained with the explicitly annotated and labeled iris images along with predefined coordinates of the bounding box. The pre-trained model is employed to adopt knowledge transfer when retrained on the standard iris datasets. Moreover, by using the weights learned during pre-training yields the quick network convergence with improved accuracy [13]. This is since the initial CNN learns generic features, such as points, blobs, and edges, and during retraining; such knowledge is successfully transferred to various diverse tasks. Since the model is not

Table 1 Summarizing literature analysis in terms of different feature extraction methods focusing on various eye regions

Category	Procedure	Merits	Limitations
Feature extraction from entire eye image	Using handcrafted feature descriptors to extract features from iris and facial images [3, 8, 25, 27, 33]	Since it doesn't require training, it can perform well with less data. Easy to implement	Requires parameters to be set empirically by experts
	A pre-trained CNN model is used to extract features from down sampled image [6]	Extract better discriminative features by self-learning from large set of images	Suffers from poor accuracy with less amount of data Huge amount of computation and processing time is required
	A shallow convolutional model is used for feature extraction and classification [35]		
	Combining handcrafted and CNN based features using certain fusion method [21–23, 37]	Enhances classification performance by integrating both self-learnt and handcrafted features	Lengthy procedure Huge computation required
	Handcrafted features are used to train lightweight CNNs [12]		
Using segmented and normalized iris region for feature extraction	Handcrafted methods are used to extract features from enhanced normalized iris image [14, 15]	Avoids additional processing over other eye regions (except iris) while feature extraction	Includes iris segmentation which suffers from poor accuracy with blur and noisy images
	CNN based feature extraction from segmented iris region [10, 16]		
	CNN based feature extraction from non-overlapped patches of normalized iris images [36]	Utilizes rich information by rigorously analyzing each local region of the normalized image	It doesn't consider the important textural details in iris and pupil nearby areas while extracting features
Feature extraction from iris region as RoI	Features are extracted from non-overlapping patches of RoI using handcrafted feature descriptors [3, 7]	Focuses on each local region for identifying key intensified pixels, i.e., discriminative features. Each patch is considered as an image where filters are employed	Large processing time due to employ ordinary filters repeatedly on multiple patches and then combining results from each patch
	Features are extracted using a pre-trained CNN model [17]		
	A YOLO model is employed for detection and classification in parallel [11]	The iris region localization and presentation attack prediction is accomplished by a sole model in single stage	Requires huge amount of images and ground truth vector to successfully perform regression and classification as a whole
Feature extraction from local and global iris regions	Handcrafted feature extraction from iris and sclera regions and summarize them [31]	Features are constituted in relatively large quantity as it applies feature extraction from local regions and then entire eye image	Additional computation is required as the same feature extraction procedure is employed repeatedly on each local region and entire image as well
	CNN based feature extraction from inner and outer regions of iris and combine them using some fusion method [38]		

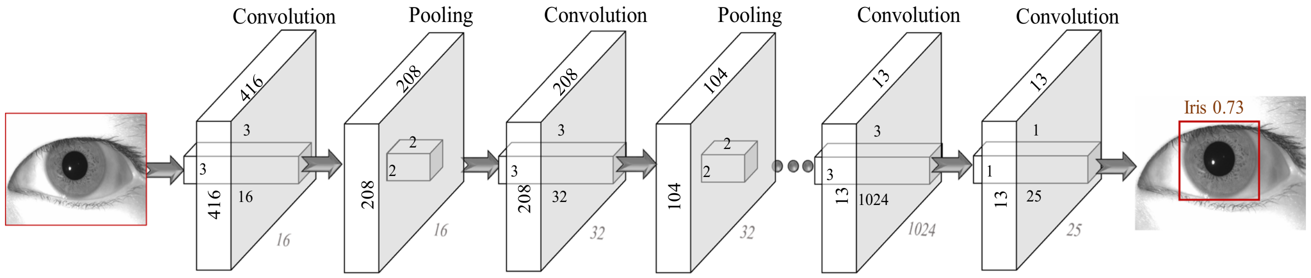


Fig. 3 Depiction of the CNN framework employed for region of interest (RoI) detection, working as a regressor, predicting the dimensions of bounding box, which tightly bounds the iris region

Table 2 Various parameters involved in the YOLO framework in the experimental setup

Layers	No of Filters	Filter Size	Stride	Output size
Conv_1	16	(3,3)	1	(416,416,16)
Max-pool	–	(2,2)	2	(208,208,16)
Conv_2	32	(3,3)	1	(208,208,32)
Max-pool	–	(2,2)	2	(104,104,32)
Conv_3	64	(3,3)	1	(104,104,64)
Max-pool	–	(2,2)	2	(52,52,64)
Conv_4	128	(3,3)	1	(52,52,128)
Max-pool	–	(2,2)	2	(26,26,128)
Conv_5	256	(3,3)	1	(26,26,256)
Max-pool	–	(2,2)	2	(13,13,256)
Conv_6	512	(3,3)	1	(13,13,512)
Max-pool	–	(2,2)	1	(13,13,512)
Conv_7	1024	(3,3)	1	(13,13,1024)
Conv_8	1024	(3,3)	1	(13,13,1024)
Conv_9	25	(1,1)	1	(13,13,25)
Softmax	–			

*The Conv_* block includes the convolution, batch normalization and max pooling operations

too deep, it is entirely fine-tuned rather than freezing some initial layers and fine-tuning the rest. Next, to further enhance the training size and to address overfitting, some data augmentation methods are employed to perform transformations such as random flipping, shearing, rotation, and cropping on the training batches. The augmented images are supplied to the model by ImageDataGenerator, an open-source tool provided by Keras [40], during model training only. Since the model does not perform classification, the classification loss is excluded from the loss function, which contains localization loss and confidence loss. Therefore, the loss function to accomplish regression (predicting the bounding box parameters), is defined as follows:

$$\begin{aligned} L = {} & \lambda_{\text{cord}} \sum_{i=0}^{C^2} \sum_{j=0}^{N} \left[B_{ij}^{\text{iris}} (x_{\text{trgt},i} - \widehat{x_{\text{pred,i}}})^2 + (y_{\text{trgt,i}} - \overset{\wedge}{\underset{\text{pred,i}}{y}})^2 \right] \\ & + \lambda_{\text{cord}} \sum_{i=0}^{C^2} \sum_{j=0}^{N} \left[B_{ij}^{\text{iris}} (w_{\text{trgt,i}} - \widehat{w_{\text{pred,i}}})^2 + (h_{\text{trgt,i}} - \widehat{h_{\text{pred,i}}})^2 \right] \\ & + \lambda_{\text{iris}} \sum_{i=0}^{C^2} \sum_{j=0}^{N} \left(B_{ij}^{\text{iris}} (O_{\text{trgt,i}} - \widehat{O_{\text{pred,i}}})^2 \right) \\ & + \lambda_{\text{no\_iris}} \sum_{i=0}^{C^2} \sum_{j=0}^{N} \left(B_{ij}^{\text{no\_iris}} (O_{\text{trgt,i}} - \widehat{O_{\text{pred,i}}})^2 \right) \end{aligned} \tag{1}$$

Here, B_{ij}^{iris} represents the case if iris exists in the ith cell and B_{ij}^{iris} signifies if the jth bounding box in the ith cell primarily contributes to the prediction. In contrast, $B_{ij}^{\text{no\_iris}}$ denotes the case when the jth bounding box in the ith cell does not contain the iris. C^2 denotes the total number of cells present in the last feature map, and N shows the number of bounding boxes to predict. In the experiment, the values for N and C are set to 5 and 13, respectively. Ultimately, the total number of bounding boxes to predict for an image is given by $C \times C{\times}N$, i.e., $13 \times 13 \times 5$. The actual values for output class O_i is computed as follows:

$$O_{\text{trgt},i} = \begin{cases} 1 & \mathit{if}\ B_{ij}^{\text{iris}} = 1 \\ 0 & \mathit{if}\ B_{ij}^{\text{no\_iris}} = 1 \end{cases} \tag{2}$$

Fundamentally, the above derivation calculates the difference in the actual and predicted values then measures the L_2 loss. λ_{coord},λ_{iris}, and $\lambda_{\text{no\_iris}}$ are the hyper-parameters employed to weight the distinct regression losses. In our experiment, these values are considered as 1, 5, and 1, respectively.

Refer to Eq. (1), the first two terms of the loss function describe the coordinates of the predicted bounding box's center, whereas the second term relates to the box's height and width. The third and fourth terms focus on the probability of the box to encompass the iris. Indeed, all the loss terms are summed together to form the unified L2 regression loss. Further, the model is trained by using the SGD with momentum, with 64 batch size. Consider that the box_{tg} and box_{pd} represent the target and predicted bounding box, respectively. Then,

$$\begin{aligned} \text{box}_{\text{tg}} &= (x_{\text{trgt}}, y_{\text{trgt}}, w_{\text{trgt}}, h_{\text{trgt}}) \\ \text{box}_{\text{pd}} &= (\hat{x}_{\text{pred}}, \hat{y}_{\text{pred}}, \hat{w}_{\text{pred}}, \hat{h}_{\text{pred}}) \end{aligned} \tag{3}$$

Here, the tuple $(x_{\text{trgt}}, y_{\text{trgt}}, w_{\text{trgt}}, h_{\text{trgt}})$ implies the target output values from the coordinates of labeled bounding boxes, whereas $(\widehat{x}_{\text{pred}}, \widehat{y}_{\text{pred}}, \widehat{w}_{\text{pred}}, \widehat{h}_{\text{pred}})$ denotes the predicted coordinate values for the bounding boxes. Let (b_x, b_y, b_w, b_h) are the outcomes of the last convolutional layer, then they are transformed through the pre-specified anchor locations (l_w, l_h) to the offsets as given below [13].

$$\begin{aligned} \widehat{x}_{\text{pred}} &= \sigma(b_x) + c_x \\ \widehat{y}_{\text{pred}} &= \sigma(b_y) + c_y \\ \widehat{w}_{\text{pred}} &= l_w exp\,(b_w) \\ \widehat{h}_{\text{pred}} &= l_h exp\,(b_h) \end{aligned} \tag{4}$$

Here, (c_x, c_y) signifies the coordinates of the upper left corner from the current cell of the resultant feature map. The default anchor locations given in [13] are used in the model, i.e., {(3.42, 4.41), (1.08, 1.19), (9.42, 5.11), (6.63, 11.38), (16.62, 10.52)}.

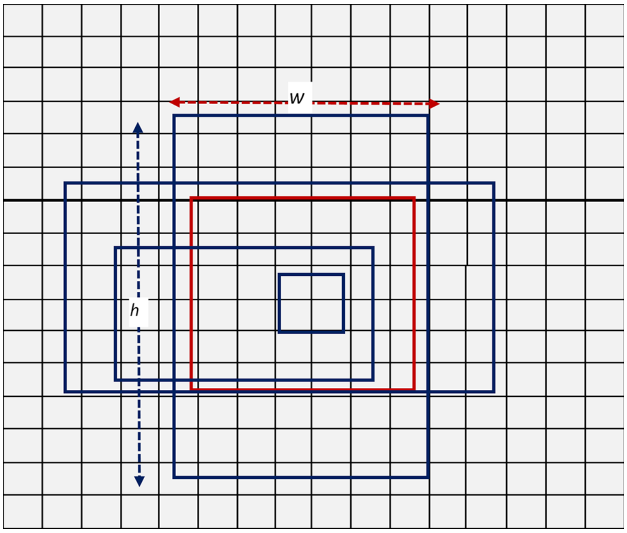


Fig. 4 The pictorial representation of the process of bounding box prediction using the pre-specified anchors

$$\text{Intersection of Union (IoU)} = \frac{\cap(\text{Box}_{\text{tg}}, \text{Box}_{\text{pd}})}{\cup(\text{Box}_{\text{tg}}, \text{Box}_{\text{pd}})} \quad (5)$$

The procedure to predict bounding boxes by incorporating the pre-specified anchors is depicted in Fig. 4. The predicted bounding boxes are shown in blue rectangles, computed from the five pre-specified anchors. This framework convolves on all grid cells and calculates the *IoU* between the target bounding box (box_{tg}) and the predicted bounding box (box_{pd}) as given in (5), and the largest IoU is observed. If the largest IoU is greater than a preset threshold, then the respective cell produces zero loss to calculate the probability of the bounding box enclosing the iris. The output of the iris localization framework corresponding to images from different datasets is shown in Fig. 5.

3.2 Selective image enhancement on RoI

The localized RoI (iris region) is cropped from the image based on the rectangle box using OpenCV in python and undergoes selective image enhancement through rescaling, sharpening, color, and contrast variation, etc. It increases the subjective and textural quality of the image to enrich the textural details. The cropped iris image is first rescaled by a factor of 1.25, and then image sharpening is performed. It highlights the fascinating minutiae in the region to eliminate the noise and to make the image more alluring. Indeed, the edge sharpening and fine details are determined by the sharp conversions in the image intensity. Further, the sharpening is produced by preserving the high-frequency modules and discarding the low-frequency components. Besides, the contrast is twisted by the variance in the illumination reflected from two neighboring surfaces. In specific, contrast refers to the distinction in chromic properties that enables an object discernable from other objects and background as well. It is obtained by the difference in the brightness and color of an object from others. There are various algorithms and linear and nonlinear functions to accomplish contrast enhancement; however, logarithmic transformation is used in our experiments. Figure 6 represents the output image samples after performing image enhancement.

3.3 Feature extraction from RoI

Feature extraction from the RoI is carried out using three different approaches: key point-based, local and global descriptors, and a deep learning-based feature extractor [3]. The key point-based feature extractors include scale-invariant feature transform (SIFT) to extract the set of local key points. Besides, the local descriptors such as LBP [25], CoA LBP [28], Multiscale BSIF (MBSIF) [41], Zernike moment [37] and Haralick features [22] perform textural analysis to constitute discriminative patterns and generate output feature-vectors. Further, the VGG-8 model is employed to extract deep learning-based features, which is an imitation of VGGNet [42] with eight layers instead of sixteen. Such feature extractors are described in the following subsections.

Key-point-based feature extraction: key points or key features denote the points in the image, which are invariant to chromatic deviations and image rescaling. In our experiment, the SIFT descriptor is used to identify unique key features from the RoI. Such key points have diverse colors, which are utilized to indicate discernment between

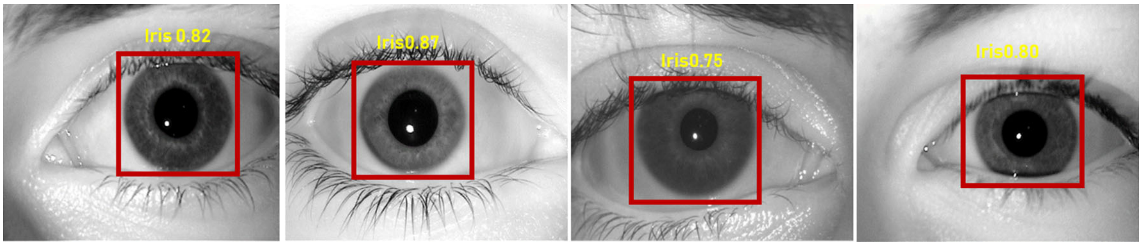


Fig. 5 Output of the iris localization framework containing the bounding box indicating the iris region along with the probability score

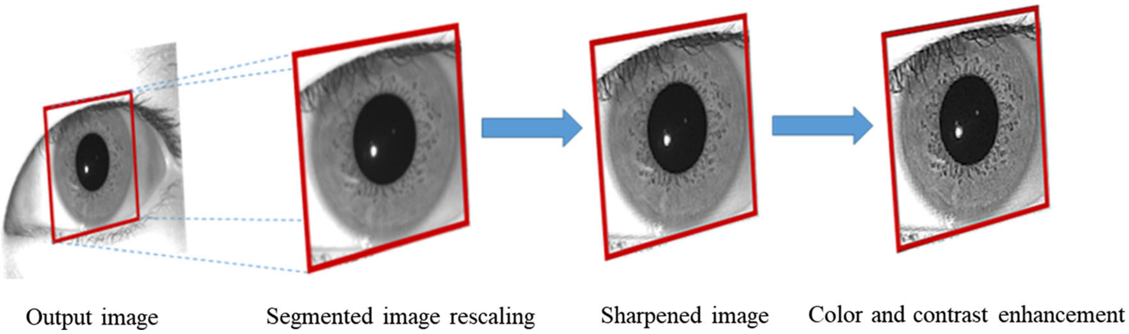


Fig. 6 Depicting image enhancement in Region of Interest (RoI) through rescaling, sharpening, color, and contras variation

key features and are considered beneficial for detecting presentation attacks with cosmetic lenses and printed iris images.

Local and global feature extraction: local descriptors refer to image neighborhood localities computed at several interest points. Usually, interest points are perceived at various measures and estimated to reprise across distinct views of an object. In addition, they are also probable to acquire the core of the object's exterior. Such feature descriptors designate the image patch surrounding the point of interest. The prime advantage of employing local features is that they have significant potential to recognize the object despite clutter and occlusion. In this work, Co-occurrence of LBP (CoA LBP), MBSIF, Zernike movement, and Haralick features are employed. CoA LBP refers to an image feature that relies upon spatial co-occurrence amongst micro-structures represented by an LBP. It was introduced to counter the limitations of conventional LBP, wherein LBP histograms, all the LBPs of micro-features, are wrapped into a sole histogram. It abandons essential information regarding spatial relations amongst the LBPs, although they may encompass information regarding the image's global structure.

Besides, MBSIF is an advancement over the traditional BSIF method, where instead of a single fixed-size filter, multiple filters with changing scales are incorporated. The multiple filter responses are combined to create an improved, unique feature set. This work utilized three domain-specific filters of sizes 17 × 17, 7 × 7, and 5 × 5 with a bit length of 12, 10, and 8, respectively, publicly provided in [43]. These filters are domain-specific since they are trained on patches of iris samples and more powerful than generic BSIF filters [41].

On the other side, the Zernike moments [37] are robust across the variations in rotation, scale, and translation, also efficaciously applied in iris segmentation and recognition [44]. In this work, Zernike moments are used to capture the fluctuations in the contour between the live iris and attack samples. An orthogonal set of polynomials is included to define the Zernike moments in an image, and a radial polynomial $P_{m,n}$ is computed over it. This radial polynomial is demarcated as follows:

$$P^n_m(\rho) = \sum_{k=0}^{\frac{m-|n|}{2}} \frac{(-1)^k \rho^{m-2k}(m-k)!}{k!\left(\frac{m+|n|}{2}-k\right)!\left(\frac{m-|n|}{2}-k\right)!} \tag{6}$$

where ρ is the difference between a point (i, j) and the center of the image, m shows the order of polynomial function and n depicts the repetition such that $|n| < m$ and $|m–n|$ is even. The Zernike function is unswervingly calculated in the Cartesian coordinate space given by:

$$Z_{m,n}(p,q) = R^n_m(\rho_{p,q})e^{-jn\theta_{p,q}} \tag{7}$$

Consider, S × S as the image size, then

$$\rho_{p,q} = \frac{1}{S} \times \sqrt{(2p-S+1)^2+(S-1-2q)^2} \tag{8}$$

$$\theta_{p,q} = tan^{-1}\left(\frac{S-1-2q}{2p-S+1}\right) \tag{9}$$

Let's consider I as an iris image, then Zernike moments are computed for (x, y) across the non-overlapping cell of $n \times n$. Several pairs of (x, y) are chosen to calculate the amplitude of multi-order Zernike moments. It aids to improve the depiction of the input image. In our experiment, Zernike moments are calculated on non-overlapped patches of 4 × 4, 8 × 8, and 16 × 16, and the resultant features are combined to produce combined feature vector.

Next, the Haralick features [22] are well-known statistical global descriptors used to encode the textural details within an image. These are effectively employed in several domains, such as medical imaging, texture classification, and face presentation attack detection. Haralick features exploit the gray level co-occurrence matrices (GLCM), a tabulation of the occurrences of distinct combinations of gray pixels in an image. Typically, the aim is to map the given unknown sample to either of a set of predefined texture classes. The textural features may be discrete histograms, scaler numbers, or empirical distributions. Moreover, such features characterize the image's textural properties such as contrast, orientation, spatial structure, and roughness, and encompass definite correlation with the target output. The GLCM represents the scatter of co-occurred pixel intensities within the image (*I*) at a well-defined pair (*Δp, Δq*) at position (*x, y*). Therefore, $\mathrm{GLCM}_{\Delta \mathrm{p},\ \Delta \mathrm{q}}$ (*x, y*) is calculated as follows:

$$\mathrm{GLCM}_{\Delta p, \Delta q}(x, y) = \sum_{p=1}^{m}\sum_{q=1}^{n}\begin{cases}1, & if\ I(p,q) = x\ ;\ I(p+\Delta p, q+\Delta q) = y\\ 0, & otherwise\end{cases} \tag{10}$$

After calculating GLCM, Haralick features are computed to encode the textural details in the image. Indeed, there are 13 different Haralick features (i.e., contrast, the sum of variance, the difference in the variance, correlation, sum of average, inverse difference moment, entropy, the sum of entropy, the sum of squares of variance, two information correlation scores, angular second moment, and difference in entropy).

After feature extraction and encoding by using the aforementioned local and global descriptors, corresponding feature-vectors are generated, and they may vary in length depending upon the number of feature values extracted by a particular descriptor. Each feature-vector is fed to a dedicated SVM classifier to generate corresponding output class, i.e., to label the given eye image as either “live” or “attack”. The training procedure of the SVM classifier with the feature-vectors to generate the output labels is described in the next subsection.

3.4 Local classification using SVM

In the proposed approach, multiple SVM classifiers are used at various stages, e.g., in best-*k* feature selection and score-level fusion. Therefore, this subsection provides a deep insight into the working principle of the SVM classifier.

Let $\{F_k\}_{k=1}^{N}$ as the training feature set, and $\{Y_k\}_{k=1}^{N}$ as the corresponding labels, the SVM attempts to learn a hyperplane Υ as follows:

$$\arg_{\gamma} \min\|\gamma\|_2^2 + \rho\sum_{k} L(\gamma, F_k, Y_k) \tag{11}$$

where ρ and *L(γ, F*$_k$*, Y*$_k$*)* denotes the penalty parameter and the loss function, respectively. Due to the efficacy of the quadratic Hinge loss in the image classification, it is employed in the experiments. The hinge loss function is expressed as follows:

$$L(\gamma, F_k, Y_k) = \left[\max\left(0,\ \gamma^T F_k Y_k - 1\right)\right]^2 \tag{12}$$

where, Y_k is set to 1 for live iris, whereas − 1 signifies spoofs. After learning γ, a test set Y_{test} is given to the SVM classifier, and the classification utilizes the sign of $\gamma^T Y_{test}$. Here, ρ is set to 0.1. As mentioned earlier, a dedicated SVM classifier is associated with each feature descriptor (selected in best-k features) for local classification, i.e., the class prediction using the single feature vector. Further, the score-level fusion of all individual classifications is accomplished to perceive the global prediction.

3.5 Global classification through score-level fusion

Let $c_1, c_2, \ldots, c_k$ represent the local outcomes of *k* classifiers, and the score-level fusion acquires a set of weights $w_1, w_2, \ldots w_k$, where $w_1 + w_2 + \cdots + w_k = 1$, to compute the fused output (C_s) as follows:

$$C_s = c_1 w_1 + c_2 w_2 + \cdots + c_k w_k \tag{13}$$

The weights $w_1, w_2, \ldots w_k$ in (13) are learned by recursively evaluating the individual performances of the classifiers on the test set. This is obtained by using varying partitioning over the train and test sets, where the train-set is assigned a large number of samples so that the classifiers may learn effective feature discernment. Though, instead of combining outputs of all classifiers, best-*k* features are identified that result in more accurate classification. In this vein, we exploit the preeminent statistical tests for the concurrent assessment of several classifiers. As all SVMs are the same, they differ in their performances due to the feature-vectors, which are used for classifiers' training. Thus, the association of each feature extractor with SVM acts as a classification algorithm, where the best-*k* methods are identified using the Friedman test [39]. It performs feature selection by simultaneously evaluating each feature with the output and ranks them based on their performances. Further, the optimal set of weights for score-level fusion is obtained by an inclusive analysis of the train-set for various experiments. In particular, the partitioning is performed as 5267 training and 1316 testing samples for IIITD CLD, 8482 training and 2120 testing samples for IIITD-CSD, 4080 training and 1020 testing samples for ND CLD, 3840 training and 960 testing samples for ND-LivDet, 6356 training and 1589 testing samples for Clarkson-2015, and 6476 training and 1619 testing samples for Clarkson-2017 datasets. Each dataset is randomly split 100 times, where the best-*k* features are identified through tenfold cross-validation on the train-set. Finally, the local responses of such *k* classifiers for the test samples are fused using (13) to obtain the global outcome. The first 50 random splits are utilized to learn weights, whereas the performance of sore-level fusion is computed using the rest 50 partitions.

1. Best-*k* Feature selection through Friedman test

In general, the Friedman test is aimed to discard a null hypothesis that given multiple classifiers are statistically similar, i.e., they all exhibit equal performances. In this vein, the Friedman test exploits two statistics expressed as follows:

$$\chi_F^2 = \frac{12D}{n(n+1)}\left[\sum_i R_i^2 - \frac{n(n+1)^2}{4}\right] \tag{14}$$

$$F_F = \frac{(D-1)\chi_F^2}{D(n-1)-\chi_F^2} \tag{15}$$

Algorithm-1: Create Multiple Image Datasets

```
Input: Image Dataset ‘D’, where $|D| = n$
Output: Three new image datasets ($v$).
        $v_i = \{v_1, v_2, v_3\}$, such that $\forall_I, v_i \not\subset D$
Procedures: Shuffle: Random permutation within the set (dataset)
    Augmentation: Transformation of the images in a certain way to increase the diversity
    Sampling: Sampling without replacement
Steps:

1. $D^{'}$ =Augmentation ($D$), such that $D \cap D^{'} = \emptyset, |D^{'}| > |D|$
2. $D^{'}$ = Shuffle (D) using the Fisher-Yates algorithm
3. $S$ = Sampling $(D,n,P,Q)$, where $P=3, Q=n/3$, $D$ =population   (dataset),
   $n$ =population size, $P$ =number of samples (3), $Q$ =size of sample ($n/3$)
   $S_i = \{S_1, S_2, S_3\}$, such that $\forall_{i,j}, S_i \cap S_j = \emptyset, i \neq j$
4. Repeat steps 2-3 for the dataset $D^{'}$.
   $S^{'}_i = \{S^{'}_1, S^{'}_2, S^{'}_3\}$, such that $\forall_{i,j}, S^{'}_i \cap S^{'}_j = \emptyset, i \neq j$
5. $T = S_i \times S^{'}_i$, $T$ is unordered
   $T = \{S_1 S^{'}_1, S_1 S^{'}_2, S_1 S^{'}_3, S_2 S^{'}_2, S_2 S^{'}_3, S_3 S^{'}_2\}$
   such that $S_i S^{'}_I = S_i \cup S^{'}_i$, consider as a single item (dataset)
6. $v$ = sampling $(T, n, P, Q)$, where $n = 6, P = 3, Q = n/6$
   i.e., the random selection of three datasets from T .
   $v_i = \{ v_1, v_2, v_3 \}$, such that $\forall_{i,j}, v_i \cap v_j = \emptyset, i \neq j$
7. Return $v$
```

where, n, R, D signify the number of datasets, average rank, and the number of classifiers, respectively. Nevertheless, in the proposed approach, the null hypothesis is modified as “all extracted features exhibit identical contribution in the output prediction.” In the Friedman test, D and n should be big enough (as a rule of thumb, $D > 10$ and $n > 5$) [39]. There are multiple feature-vectors; each with separate SVMs, acting as classifiers (n), yet the number of datasets (D) is limited. Therefore, subsampling is performed over the datasets as given in Algorithm-1 to counter it, where three distinct samples are created from each dataset. Thus, a total of 18 sampled datasets are produced from six datasets and are considered as separate datasets. However, sampling on the raw dataset seems inadequate as each iris image is not included in the sampled dataset. In addition, the size of the sampled datasets would be perilously less, and thus, the overall outcome may be affected. Therefore, before subsampling, image augmentation on each dataset is carried out by using *imagedatagenerator* (an image processing tool provided in Keras [40]). It performs certain transformations on each image matrix in the dataset (as described in the next subsection) and generates similar augmented images.

2. Image augmentation

In order to generate auxiliary training samples, several augmentation methods are incorporated that accomplish various transformations on the images of the given iris datasets. Such transformation includes rotation, flipping, shearing, shearing after rotation, rotation after shearing at varying directions and angles, etc. Indeed, such transformations are similar to regular matrix operations, i.e., the input image matrix is modified in terms of pixel values and locations surrounding the axis. Flipping is implemented horizontally as well as vertically, where pixels are moved along the height and width. Similarly, rotation moves the pixel values in the 2D plane counter-clockwise with the predefined angle about the origin. Besides, shearing amends pixel values according to the variation in their distances from all axis. These transformations are carried out by employing ‘ImageDataGenerator’, i.e., a class facilitated by Keras [40] for image pre-processing. The parameters specified for such transformations are given as: shear ($\leq$ 0.2), rotation ($\leq$ 40), flip (horizontal and vertical = True), height shift ($\leq$ 0.2), and width shift ($\leq$ 0.2). It aids in producing auxiliary images with analogous features to augment and enrich the model input.

4 Experimental framework and discussion

Aiming to realize iris liveness detection, six different iris datasets containing iris images with various attack variants, are considered in this study. The primary reason behind including many datasets is a prerequisite of the Friedman test for best-*k* feature selection. However, the proposed approach is examined on each dataset for performance validation. A series of experiments are performed on these datasets to validate the efficacy of the proposed approach. The following subsections describe the databases along with the validation protocols used in the experiments for method assessment, and the respective outcomes are also discussed. A comparative study among the proposed approach and the state of the arts is also described.

4.1 Description of iris datasets and validation protocols

Table 3 demonstrates the iris datasets used in this work, along with the underlying image distribution and attack types. The IIITD contact lens dataset (CLD) [45] contains live, soft, and patterned lens iris samples of 101 distinct subjects that are captured through Cogent and Vista sensors. Besides, IIITD combined spoofing dataset (CSD) [37] contains live, patterned, print-scan, and print-capture iris images. Notice that, IIITD-CSD contains the IIITD contact lens dataset as a part of it; thus, we exclude it from the combined spoofing dataset. Further, ND-LivDet-2017 [10] and ND contact lens 2013 [46] dataset contains merely live irises and patterned iris samples. Furthermore, both

Table 3 Description of iris PAD datasets in terms of image distribution with arrack types

No.	Database	Types of Images	Live	Spoof
1.	IIITD Contact Lens Iris Database [45]	Soft and Patterned contact lens, live	4310	2273
2.	IIITD Combined Spoofing Database [37]	Patterned contact lens, Print + capture, Print + scan, live	6022	4580
3.	ND-Iris Contact lens 2013 [46]	Patterned contact lens, live	3400	1700
4.	ND-LivDet-2017 [10]	Patterned contact lens, live	2400	2400
5.	LivDet-Iris-2015 Clarkson [10]	Patterned contact lens, Print + scan, live	1906	6039
6.	LivDet-Iris-2017 Clarkson [10]	Patterned contact lens, Print + scan, live	3954	4141

LivDet-Iris 2015 and LivDet-Iris 2017 (Clarkson) [10] datasets contain images of live irises, iris printouts, and textured contact lenses. In addition, a "Combined" dataset is also prepared for some experiments by merging images from all datasets. Each dataset is provided with train and test partitions to facilitate the training and testing of algorithms. The datasets belonging to LivDet-Iris 2017 further divide the test samples into two groups; test-known and test-unknown. In the former group, both live and artifacts possess the same "known" properties like train samples. Besides, the second group has unknown or different properties than train samples. Note that this work adheres to binary classification, i.e., live versus attack, and does not discriminate among attack types.

For all the abovementioned datasets, the experiments follow the predefined train-test partitioning for feature extraction and SVM training. However, fivefold cross-validation is used for best-*k* feature selection, where each dataset is divided into five equal parts. Afterward, in each training phase, one part is considered as test set, and the rest four are used for algorithm training. Notice that the fivefold cross-validation is employed in merely the Friedman test to compute area under the curve (AUC) values, whereas the remaining experiments follow the procedure described in subsection-III(D), i.e. "score-level fusion." The feature extraction methods constitute the discriminative features from the train and test sets and constitute the train and test features. The classifiers are trained using train-features to realize binary classification, where authentic iris samples are labeled as "live" and artifacts denote "attack." In the experiments, the PAD performance is expressed as per ISO/IEC SC37 [47] metrics as below:

- Accuracy: Ratio of correctly classified samples out of total samples.
- Bona fide presentation classification error rate (BPCER): Ratio of live irises, incorrectly classified as attacks, out of total samples.
- Attack presentation classification error rate (APCER): Ratio of attack samples erroneously classified as live, out of total samples.
- Average classification error rate (ACER): Average of BPCER and APCER.
- Equal error rate (EER): The point/value, where ACPER and BPCER are equal.

Here, APCER and BPCER correspond to false acceptance rate (FAR) and false rejection rate (FRR), respectively. Such error rates vary based on the variation in the threshold on the classifier's output. As mentioned above, the point/value, where both error rates become equal, is referred to as an equal error rate (EER). The trade-off between FAR and FRR is outlined using detection error trade-off (DET) curves based on varying thresholds [47], where the diagonal line aids in EER computation with the point, where this line meets the DET curve.

4.2 Best-k feature selection

Table 4 compares seven distinct algorithms (feature extraction plus SVM) on 18 datasets using the Friedman test, where the intermediate results are expressed in terms of average ranks. Besides, separate ranking is done according to the higher values of the area under the curve (AUC) for each dataset as in [39]. The fivefold cross-validation is employed on each dataset, and an average AUC is computed. Although there are several feature extraction algorithms available in the literature, few selected methods are examined due to their improved performances reported in previous works [21, 22, 37]. In specific, SIFT, LBP, MBSIF, CoA LBP, Haralick features, Zernike moments, and VGG-8 model exhibit better performances in textural classification. In this view, these methods are embedded with dedicated SVM classifiers to discriminate between attack and live samples and examined 18 different sampled datasets. More details about the Friedman test can be found in [39].

Table 4 Analysis of seven distinct features on 18 subsampled iris PAD datasets for best-k feature selection through the Friedman test

Datasets	SIFT	CoA LBP	MBSIF	Zernike	LBPV	Haralick	VGG-8
IIITD-s1	0.945 (2)	0.681(6)	0.952(1)	0.794(5)	0.842 (4)	0.763 (7)	0.927(3)
IIITD-s2	0.957(1)	0.794(5.5)	0.932(2)	0.799(5.5)	0.821(4)	0.731(7)	0.927(3)
IIITD-s3	0.951(1)	0.716(6)	0.926 (2.5)	0.683(7)	0.793(4)	0.746(5)	0.921(2.5)
IIITD-CSD-s1	0.962 (1.5)	0.813(4)	0.891 (3)	0.723 (6)	0.799 (5)	0.687 (7)	0.962 (1.5)
IIITD-CSD-s2	0.913 (2.5)	0.746 (5)	0.917(2.5)	0.706 (6.5)	0.810 (4)	0.706 (6.5)	0.922 (1)
IIITD-CSD-s3	0.924 (3)	0.723 (6)	0.945 (1)	0.744 (5)	0.832 (4)	0.690 (7)	0.933 (2)
ND CLD-s1	0.935 (1)	0.802 (4)	0.895 (3)	0.703 (7)	0.786 (5)	0.726 (6)	0.924 (2)
ND CLD-s2	0.947 (1.5)	0.744 (5)	0.899 (3)	0.721 (6.5)	0.771 (4)	0.728 (6.5)	0.947 (1.5)
ND CLD-s3	0.910 (3)	0.717 (5)	0.937 (1.5)	0.702 (6)	0.846 (4)	0.679 (7)	0.931 (1.5)
ND-LivDet-s1	0.926 (2)	0.723 (7)	0.852 (4)	0.790 (5)	0.861 (3)	0.764 (6)	0.942 (1)
ND-LivDet-s2	0.929 (1.5)	0.791 (4)	0.927 (1.5)	0.751 (6)	0.780 (5)	0.712 (7)	0.911 (3)
ND-LivDet-s3	0.942 (1.5)	0.812 (4)	0.940 (1.5)	0.721 (7)	0.801 (5)	0.761 (6)	0.929 (3)
Clarkson15-s1	0.891 (3)	0.771 (4.5)	0.901(2)	0.737(6)	0.776 (4.5)	0.682 (7)	0.923 (1)
Clarkson15-s2	0.889 (3)	0.782 (5)	0.901 (2)	0.754 (6)	0.811 (4)	0.707 (7)	0.941 (1)
Clarkson15-s3	0.921 (2.5)	0.792 (4.5)	0.923 (2.5)	0.724 (6)	0.799 (4.5)	0.703 (7)	0.944 (1)
Clarkson17-s1	0.917 (1)	0.712 (6)	0.791 (4)	0.740 (5)	0.802 (3)	0.691 (7)	0.901 (2)
Clarkson17-s2	0.930 (1)	0.763 (4.5)	0.902 (2.5)	0.767 (4.5)	0.752 (6)	0.740 (7)	0.906 (2.5)
Clarkson17-s3	0.921 (1.5)	0.811 (4)	0.921 (1.5)	0.752 (6)	0.792 (5)	0.732 (7)	0.901 (3)
Average rank	**1.861**	**5.000**	**2.277**	**5.888**	**4.333**	**6.666**	**1.972**

The Ranks in the brackets are assigned based on the ascending values of AUCs reported for individual datasets

*s1, s2, s3 refer to sampled datasets generated from the proposed sampling algorithm Algorithm-1

Considering Table 4, χ_F^2 and F_F are evaluated as 91.29 and 86.70 using (14) and (15). F_F is distributed with (7–1) and (7–1) (18–1) degrees of freedom based on F distribution for 7 classifiers and 18 datasets. Since the critical value of F (6, 102) at $\alpha = 0.05$ is $2.14 \ll F_F$; thus, we reject the null hypothesis by following the Friedman statistics. Further, it can be concluded from the average ranks that SIFT, VGG-8, and MBSIF are top three; whereas, CoA LBP, Zernike, and Haralick features are bottom three algorithms. It is worth identifying their statistical difference to find best-k features, which is carried out by using two post hoc tests (Nemenyi test and Bonferroni Dunn test) [39]. These tests state that two classifiers statistically differ if the difference in their average ranks is larger than the critical difference (CD), which is computed as follows:

$$\mathrm{CD} = q_\alpha \sqrt{\frac{n(n+1)}{6D}} \tag{16}$$

Here, q_α is the critical value for given α. Next step is to employ the Nemeny test for 7 classifiers, where $q_\alpha = 2.949$ (for $\alpha = 0.05$), and CD = 2.123. It infers that SIFT, VGG-8, and MBSIF perform equally. Likewise, Zernike, CoA LBP, and Haralick features are similar. Though, nothing is decided for LBPV, as its average ranks differ from that of MBSIF and CoA LBP methods by less than the critical difference. Further, at $\alpha = 0.1$, $q_\alpha = 2.693$, the CD is computed as 1.939, which signifies that LBPV is significantly different from MBSIF. Thus, LBPV is not included in best-k features. Consequently, SIFT, VGG-8, and MBSIF are selected as best-k features.

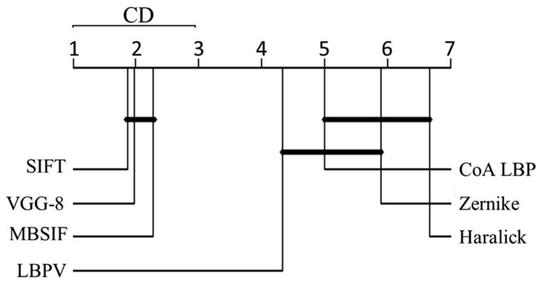


Fig. 7 Graphical illustration of results reported by Friedman test and Nemeny test, where SIFT, VGG-8, and MBSIF are ranked in top three features, and are statistically similar

Figure 7 illustrates the outcome of the post hoc tests, where it is perceived that the average ranks also demonstrate a fair assessment of the classifiers. Besides, the top horizontal line within the figure depicts the axis to plot the average ranks from left (lowest value/finest rank) to the right (higher value/lowest rank); thus, the methods on the left are superior. Besides, the methods that are statistically similar are connected via a horizontal line. The result unveils that SVMs report three best classifications with

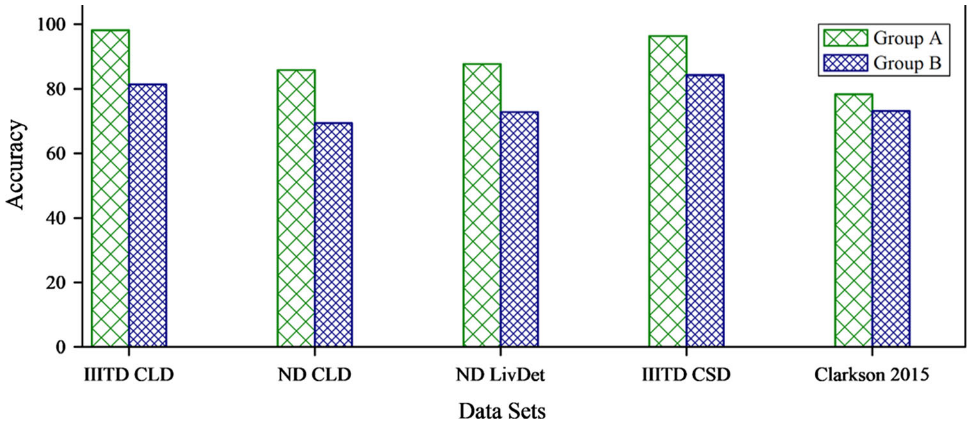


Fig. 8 Performance of two different groups of features (group-A and group-B) clustered by Friedman test, on five datasets. The feature extraction methods plus SVMs (in each group) are individually trained and the results are computed using score-level fusion on their individual scores. It is observed that methods in group-A perform significantly better for all datasets

SIFT, VGG-8, and MBSIF features. Therefore, these three features are combined through score-level fusion expressed in (13) to obtain the final output.

4.3 Validation of Freidman test's outcome

As depicted in Fig. 7, the Friedman test clustered the features into two groups according to the similarity in their performance on various databases. We consider these as group-A (SIFT, MBSIF, VGG-8) and group-B (LBPV, CoA LBP, Zernike, Haralick). As discussed in Section-B, group-A is selected as the most discriminative (optimal) feature set by the Nemeny test. This subsection further validates the effectiveness of such a feature set through a comparative analysis between group-A and group-B. To achieve this, methods in both groups are examined on all six original datasets; IIITD CLD, ND CLD, ND-LivDet, IIITD-CSD, and Clarkson. The training and testing procedures follow the validation protocols provided with each dataset, i.e., validation is performed on the predefined test sets. More in detail, methods in each group individually perform feature selection and classification, whereas the final score is obtained through performing score-level fusion on their outcomes. The experimental results for all datasets are shown in Fig. 8, where methods in group-A outperform group-B with significant performance improvement over all datasets. Here, one question arises "why the selected k features are optimal?" The reason is the underlying feature extraction procedures of MBSIF, SIFT, and VGG-8 methods. In specific, the MBSIF method uses domain-specific filters [43] to construct iris features which are more powerful than the generic filters [8]. Similarly, VGG-8 is retrained on iris datasets, and thus, the feature maps learned by the VGG-8 model are also domain-specific. Therefore, their combination yields a significant improvement in the iris pattern discrimination. Besides, SIFT features identify key points within the iris region that certainly differs for presentation attacks with cosmetic lenses and printed iris images [25]. Each of these methods individually performs better classification. However, their score-level fusion causes an additional improvement towards the correct output prediction.

4.4 Fusion methods comparative analysis

In this study, one probable question arises that why to use "score-level fusion" instead of others. To answer this, we demonstrate a fair comparison among four distinct yet widely used fusion methods, i.e., score-level fusion, majority voting, feature-level fusion, and rank-level fusion. Except for feature-level fusion, all methods work at the classifier-level, i.e., on the predicted output labels. Whereas, in feature-level fusion, all three features of iris images are concatenated to form a combined feature-vector that is fed to the SVM classifier for output prediction. Table 5 summarizes the outcomes of the abovementioned fusion methods on all datasets in terms of accuracy, APCER, BPCER, and ACER. Although there is no universal trend towards using an explicit fusion method, score-level fusion is used comparatively more in iris related literature. Additionally, in the experimental outcomes, score-level fusion outperforms other counterparts with a significant margin. The reason behind score-level fusion performing better is that it considers the fair contribution of each feature in the output prediction based on their prediction accuracy. More in detail, instead of neglecting the features upon wrong prediction (as in majority voting), each feature is assigned some weights, and thus a fair contribution from each is achieved.

4.5 Intra-domain evaluation

In this validation scheme, the proposed approach is evaluated on individual datasets, where the algorithm's training and testing are carried out on the predefined train-test partitions within each dataset. However, an additional sub-splitting is created during training, where 20% of images are selected randomly from each train-set to serve as the

Table 5 Results of four different fusion approaches on various datasets

Dataset	Feature-level fusion				Majority voting				Rank-level fusion				Score-level fusion			
	Accuracy	APCER	BPCER	ACER	Accuracy	APCER	BPCER	ACER	Accuracy	APCER	BPCER	ACER	Accuracy	APCER	BPCER	ACER
IIITD CLD	85.31	14.63	3.57	9.10	91.34	6.31	1.92	4.11	94.53	3.15	1.39	2.27	99.87	0.00	0.54	0.27
ND CLD	86.92	17.70	0.93	9.30	89.72	7.29	2.96	5.12	94.67	4.62	0.89	2.75	98.21	0.08	1.19	0.64
ND-LivDet	81.12	11.21	3.67	7.44	88.71	9.21	2.05	5.63	92.11	6.42	2.91	4.66	96.73	2.02	2.93	2.47
IIITD-CSD	91.06	7.66	4.37	6.01	89.44	9.68	0.93	5.30	93.85	4.67	1.99	3.33	97.13	0.72	1.81	1.26
Clarkson 2015	83.16	20.19	2.75	11.47	84.63	14.71	4.31	9.51	88.33	12.71	4.03	8.37	97.74	2.92	1.12	2.02
Clarkson 2017	84.80	18.54	1.92	10.23	83.14	21.22	7.81	14.51	87.29	10.32	8.71	9.51	97.96	2.43	0.79	1.61

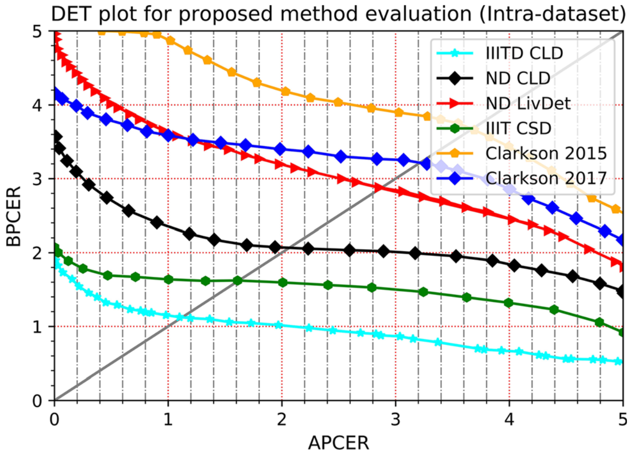


Fig. 9 DET plots for Intra-sensor evaluation of the proposed method on various datasets. The method performs best for IIITD CLD dataset, while highest misclassification rate is reported for IIITD-CSD dataset

validation set. In a nutshell, the selected feature extractors, i.e., SIFT, MBSIF, and VGG-8 extract features that are fed to dedicated SVMs to map the given iris images to either real or attack category and the outcome is obtained after score-level fusion. Afterward, two vectors for the error rates (APCER and BPCER) are calculated based on the varying threshold on the SVM's output. Further, DET curves are plotted for each dataset that plots APCER against BPCER, as shown in Fig. 9. The EER value for the dataset is obtained by observing the point on the diagonal line where the curve intersects and is listed in Table 6.

It is observed from the intra-dataset evaluation results that the proposed approach works significantly well while trained and examined on the sole dataset. The discrimination error rate is reduced up to 1.07% and 1.62%, respectively, for IIITD Contact Lens and IIITD Combined Spoofing datasets. It infers that the attack patterns with textured and printed iris can be successfully discriminated from genuine samples. Besides, the proposed method performs PAD with less than 3% misclassification rate for ND datasets. However, for Clarkson datasets, there is still a requirement to further diminish the misclassification error.

4.6 Cross-domain evaluation

Except Clarkson 2017, all remaining datasets facilitate cross-sensor evaluation, since images in their train and test partitions are captured through distinct sensors and environments. Therefore, in this section, we attempt to explore the proposed method in cross-domain, where the train and test sets possess high intra-class variations in the iris samples. First, the inter-domain evaluation is performed at the sensor-level, where different sensor images within the same dataset are served as train-test sets. As both the IIITD datasets (CLD and CSD) contain images captured through Cogent and Vista sensors, we design train-test pairs from these sensors for both datasets. Likewise, images in ND datasets (ND CLD and ND-LivDet) were captured from IrisGuard AD100 (ND-I) and LG4000 (ND-II) sensors; thus these are also arranged in cross-sensor train-test pair. The performance outcome of the proposed approach for these sensor pairs in terms of accuracy and error rates is depicted in Table 7, and the respective DET curves are shown in Fig. 10.

Further, images in Clarkson 2015 were captured using Dalsa and LG sensors, where the textural details within the respective images, along with the acquisition pattern, differ significantly. Therefore, it is expected that the error rates would be comparatively high for LG and Dalsa train-test pair. The DET curve corresponding to the Clarkson 15 dataset is illustrated in Fig. 11. It can be observed from the experimental outcomes that the proposed approach results in better accuracy for Cogent → Vista, ND-I → ND-II pairs. However, for LG-Dalsa sensor pairs, it exhibits a comparatively high misclassification error rate.

In the next phase, different datasets (cross-dataset) are considered as train-test pairs to validate the likelihood of transfer learning through these datasets. Since IIITD CLD and IIITD-CSD, both datasets contain iris samples captured with Vista and Cogent sensors; thus, the intra-class variation exists at sensor-level instead of dataset-level. Therefore, we may expect that direct cross-dataset evaluation would perform similar to cross-sensor. Accordingly, the cross-dataset evaluation experiment is conducted, where training and testing are performed over IIITD-CSD and IIITD CLD datasets, respectively. On the other side, ND CLD and ND-LivDet datasets also contain images captured using identical sensors; thus, knowledge transfer may be expected. However, according to the results shown in Table 8 and Fig. 12, the proposed method doesn't generalize pretty well for cross-dataset validation. The reported EER values for IIITD and ND datasets are 16.07% and 22.10%, respectively. On the other side, the texture of Clarkson samples differs from other datasets to a huge extent, as can be seen in Fig. 13. The combination of Clarkson with any other dataset as a train-test pair results in accuracy analogous to random predictions. The results exhibit that the variation in the datasets in properties (textured lens, print-capture, print-scan, or a mix of both), arrangement (concerning sizes of distribution among classes), acquisition sensors, and environmental conditions limit the ability to knowledge transfer.

It may be inferred from Table 8 that cross-dataset evaluation is not efficacious. However, a further possibility to obtain a successful evaluation in cross-domain is to pool images of all datasets at one place to create combined-dataset. The pooling is done according to correspondence in the individual train-test partitions. More in detail, the images in the training and testing partitions are pooled separately. In Clarkson 2017 and 2015 datasets, the test-unknown and test-known partitions are retained while dataset pooling, which is not concerned with other databases. In this regard, two different test partitions are generated "known-test," which contains the test-known partition from Clarkson and test sets of all other datasets, and "unknown-test", which contains the test-unknown partitions of both Clarkson datasets. Table 9 shows the proposed method's performance over the combined-dataset while testing with both the aforementioned test sets in terms of ACER %. With known prediction, the proposed method achieves 5.81% ACER with classification accuracy higher than 90%. Besides, with the unknown prediction, the proposed method successfully reduced ACER to 7.22%, with classification accuracy more than 88.8%. The corresponding DET curves are depicted in Fig. 14.

Table 6 Error rates resulted by the proposed approach within intra-dataset domain

Datasets	EER (%)
IIITD CLD	1.07
ND CLD	2.02
ND-LivDet	2.92
IIITD-CSD	1.62
Clarkson 2015	3.85
Clarkson 2017	3.25

4.7 Comparative analysis with state of the arts

This section demonstrates a comparative study between the proposed method and the state of the arts performing best over the datasets used in this study. As ND-LivDet and Clarkson datasets belong to LivDet-Iris 2017 competition, their corresponding results are compared with the

Table 7 Experimental results reported by the proposed approach in cross-sensor evaluation

Cross-sensor pair	Accuracy	APCER	BPCER	ACER	EER
IIITD-Congent → IIITD-Vista	99.88	3.22	0.19	1.70	2.86
IIITD-Vista → IIITD-Congent	98.75	4.08	2.15	3.11	3.09
ND-I → ND-II	98.21	2.93	1.36	2.14	1.87
ND-II → ND-I	97.53	4.79	3.91	3.25	4.68
LG → Dalsa	86.31	7.89	4.25	6.07	5.93
Dalsa → LG	82.54	9.01	3.92	6.46	7.19

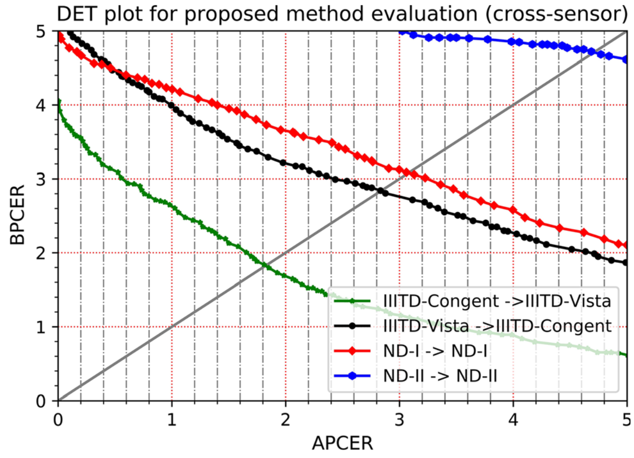


Fig. 10 DET plots for cross-sensor evaluation of proposed method on various datasets. The method performs best for Congent- > Vista pair, while highest misclassification rate is reported for ND-II- > ND-I

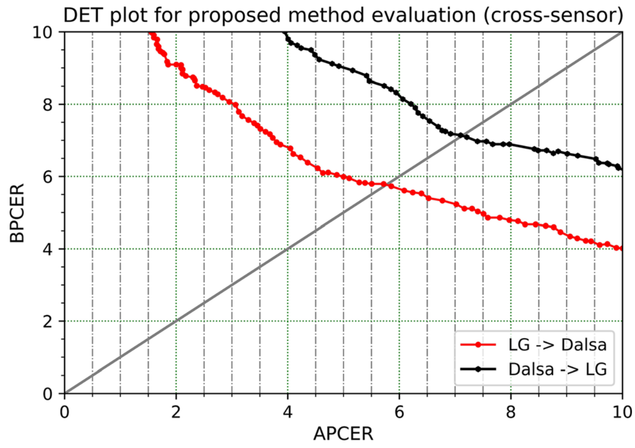


Fig. 11 DET plots for cross-sensor evaluation of proposed method on Clarkson 2015 dataset. The method performs better while using LG and Dalsa sensor images for training and testing, respectively

competition winner along with a recently introduced PAD method based on Meta-Fusion [12]. However, for other datasets, the results are compared with another state of the arts performing best on the respective datasets. Moreover, we have carefully implemented a DensePAD framework as described in [48] on all the datasets considered in this work. This is because the DensePAD framework has not been examined on these datasets in the original work. Table 10 summarizes the results from the state of the arts (according to the dataset used) and compares it with the proposed method in terms of error reduction. The term "error reduction" refers to the % of error reduced by the proposed method compared to the state of the arts. It is observed that except for the ND-LivDet, the proposed approach outperforms the state of the arts for all datasets. For the IIITD CLD dataset, the proposed method achieves more than a 40% error reduction. With the ND CLD dataset also, it successfully reduces the error rate by 17.73%. However, in the case of Clarkson, there is no significant error reduction reported. Besides, for the ND-LivDet dataset, the proposed method lacks 69.17% from the existing counterpart. The DET curves resulted from the proposed method corresponding to the abovementioned experiments are represented in Fig. 15.

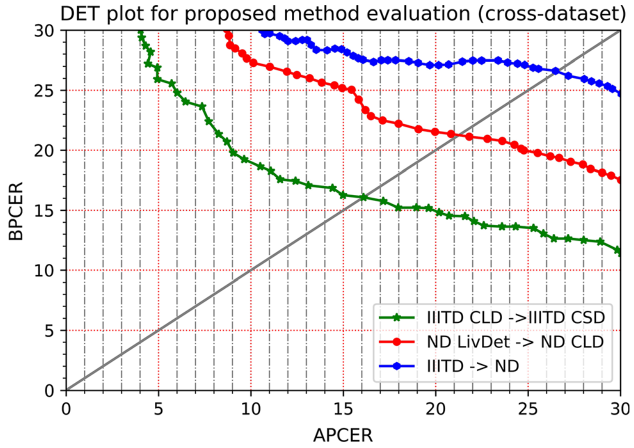


Fig. 12 DET plots for cross-dataset evaluation of proposed approach on three dataset pairs. It reports less discrimination accuracy for IIITD and ND datasets compared to intra-domain counterparts. Moreover, for Clarkson, the performance is not better than the random prediction and thus not included here

4.8 Discussion

The proposed scheme incorporates five subsequent steps, i.e., RoI localization, image enhancement, feature extraction, best-*k* feature selection, and classification to

Table 8 Experimental results reported by the proposed approach in cross-dataset evaluation

Cross-dataset Pair	Accuracy	APCER	BPCER	ACER	EER
IIITD-CSD → IIITD CLD	73.71	12.03	18.61	15.32	16.07
ND Liv→ ND CLD	70.21	18.23	20.08	20.15	22.10
IIITD → ND	63.43	27.02	20.24	23.63	27.64
Clarkson → *	Random prediction				

*Represent any dataset

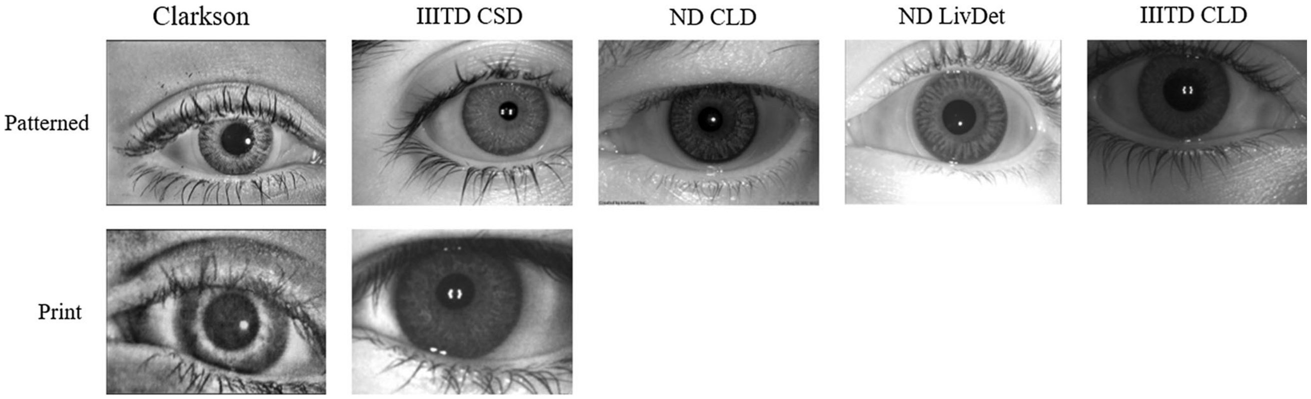


Fig. 13 Presentation attack samples from various iris datasets representing intra-class variation, where PAD approaches face difficulties

accomplish iris PAD. Each step has some significance and contribution to the attack prediction. The feature extraction procedure incorporates multiple feature extraction methods, including handcrafted and CNN-based. It is believed that CNN itself has enough potential to constitute discriminatory features from the images to perform errorless classification. However, CNN requires thousands of images per class to learn respective features that are not currently available in iris datasets. Moreover, the textural quality in images within iris datasets significantly varies due to differences in hardware and wavelength range of different iris sensors, as depicted in Fig. 13. This, in turn, results in intra-class variation in iris datasets that may not be captured through a single feature extraction method. Since each method analyzes the iris samples from a certain perspective, using multiple features tends to analyze images from multiple angles and may improve discrimination. Further, the feature selection procedure based on the Friedman test removes the redundant features with the insignificant contribution in the output prediction and results in an optimal feature set to improve iris PAD.

The experimental results infer that the proposed optimal feature set exhibits excellent performance for intra-dataset iris PAD with the least error resulted in the IIITD CLD dataset. Further, a trivial upsurge in the error rate is observed with cross-sensor deployment, yet the best outcome is observed when the train and test samples are borrowed from ND-I and ND-II datasets, respectively. Besides, the cross-dataset assessment results in a substantial increase in the error rate, where the highest two error rates have resulted in ND-LivDet → ND CLD and IIITD CLD → ND CLD as training and testing datasets, respectively. It infers that there is a huge scope towards diminishing the iris presentation attack detection errors in cross-domain setup. Furthermore, the analysis of fusion approaches to combine the features in the optimal feature set suggests that score-level fusion is an adequate choice to improve cross-domain iris PAD.

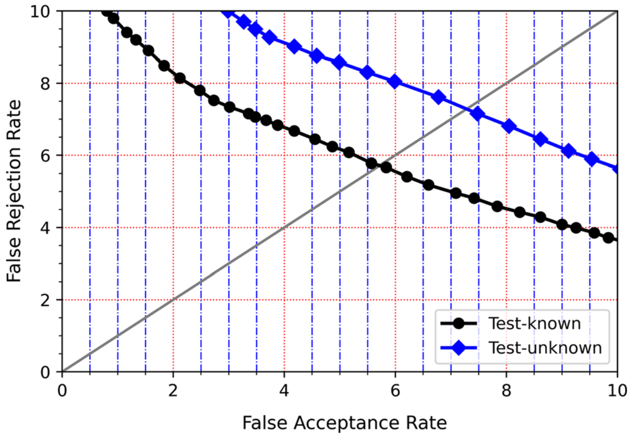


Fig. 14 DET plotes resulted by the proposed method for combined-dataset evaluation

The entire test time procedure, i.e., RoI localization, feature extraction, and SVM classification requires an average of 0.021 s for a given sample. Notice that this execution time is corresponding to a single image instead of the entire dataset. The computational platform used to execute the experiment is Intel Scalable processors Xeon 4114, 64 GB DDR4 RAM, GTX 1080Ti 11 GB GPU card.

Table 9 Experimental results reported by the proposed approach in combined-dataset evaluation

Test set	ACER %
Test-known	5.81
Test-unknown	7.22

5 Conclusion

This study focuses on improving discrimination between live iris and attack patterns/samples to enrich iris presentation attack detection. It primarily emphasizes the iris attacks that are launched at the sensor-level through a

Table 10 Comparison with state of the arts in terms of EER %

Datasets	LivDet-2017 winner [10]	Meta-fusion [12]	Other SoTA	DensePAD [48]	Proposed	Error reduction (%)
IIITD CLD	–	–	1.79 [23]	2.11	1.07	40.22
ND CLD	–	–	0.778 [38]	0.90	0.64	17.73
ND-LivDet	4.03	3.28	1.46 [22]	3.92	2.47	– 69.17
IIITD-CSD	–	–	1.30 [22]	8.135	1.26	3.07
Clarkson	9.59	9.45	10.66 [11]	11.45	9.47	0.21

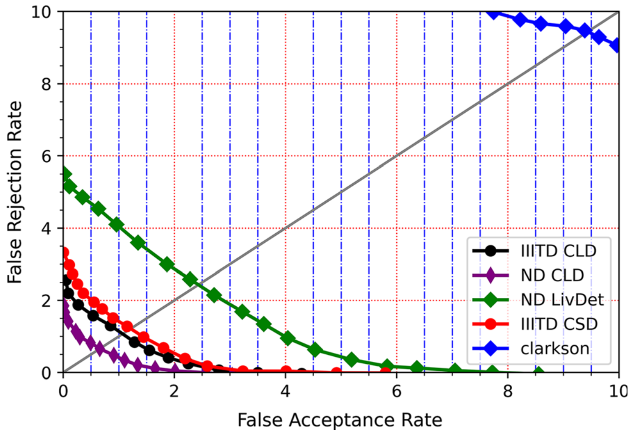


Fig. 15 DET plotes resulted by the proposed method for various datasets with predefined train and test samples

patterned contact lens, printed and scanned copies of genuine iris images. To deal with such attacks, an approach is proposed with a sequence of phases, each focusing on mitigating a certain challenge. The YOLO approach localizes the iris region without pattern loss while retaining the important textural details where discriminatory patterns exist. The RoI localization reduces the amount of computation required for feature extraction from the iris samples. The feature extraction procedure with handcrafted and CNN-based methods aimed to construct features from multiple perspectives. Further, the feature selection reduces the number of features to process without compromising the average classification accuracy. Therefore, it again yields a significant reduction in computational cost and execution time, which includes feature extraction from iris images and the corresponding classifier's training procedure. As a result of feature selection, SIFT, MBSIF, and VGG-8 features are selected as the top three features discriminating significantly among the live and attack patterns. Further, these features are combined by performing score-level fusion on the corresponding classifier's outcomes. The feature selection is robust as it is unbiased towards a certain dataset; instead, the features are examined on multiple iris PAD datasets to observe their consistency in cross-domain.

On comparing the proposed method with state of the arts, it is concluded that except for the ND-LivDet dataset, it outperforms all existing methods with significant error reduction. The improved performance is due to the efficacy of domain-specific MBSIF filters in textural feature construction, the robust key points detection through SIFT features to identify printed iris and cosmetic lenses, and the iris-specific feature maps learned by the VGG-8 model. Also, the score-level fusion boosts the accuracy by assigning appropriate weights to each feature. Although the proposed approach is lengthy, yet minimizes the misclassification error rate of both attack and genuine iris patterns.

Compliance with ethical standards

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Choudhary M, Tiwari V, Venkanna U (2019) Enhancing human iris recognition performance in unconstrained environment using ensemble of convolutional and residual deep neural network models. Soft Comput. https://doi.org/10.1007/s00500-019-04610-2
2. Czajkaand A, Bowyer KW (2018) Presentation attack detection for iris recognition: an assessment of the state of the art. ACM Comput Surv 51(4):86-1–86-35
3. Hu Y, Sirlantzis K, Howells G (2016) Iris liveness detection using regional features. Pattern Recogn Lett 82(02):242–250
4. Rigas I, Komogortsev OV (2015) Eye movement-driven defense against iris print-attacks. Pattern Recogn Lett 68(2):316–326
5. Lee EC, Park KR, Kim J (2006) Fake iris detection by using purkinje image. In: Proceedings of the international conference on advances on biometrics (ICB'06), of lecture notes in computer science. Springer, Hong Kong, vol 3832, pp 397–403
6. Choudhary M, Tiwari V, Venkanna U (2019) An approach for iris contact lens detection and classification using ensemble of customized DenseNet and SVM. Futur Gener Comput Syst 101:1259–1270

7. He Z, Sun Z, Tan T, Wei Z (2009) Efficient iris spoof detection via boosted local binary patterns. In: Proceeding of ICB, pp 1080–1090
8. Doyle JS, Bowyer KW (2015) Robust detection of textured contact lenses in iris recognition using BSIF. IEEE Access 3:1672–1683
9. Kokkinos I, Bronstein MM, Yuille A (2012) Dense scale invariant descriptors for images and surface. Research report rr-7914, INRIA
10. Yambay D et al (2017) LivDet iris 2017 Iris liveness detection competition 2017. In: Proceeding of 2017 IEEE international joint conference on biometrics (IJCB), Denver, CO, pp 733–741
11. Chen C, Ross A (2018) A multi-task convolutional neural network for joint iris detection and presentation attack detection. In: Proceeding of IEEE winter applications of computer vision workshops (WACVW), Lake Tahoe, NV, pp 44–51
12. Kuehlkamp A, Pinto A, Rocha A, Bowyer KW, Czajka A (2018) Ensemble of multi-view learning classifiers for cross-domain iris presentation attack detection. IEEE Trans Inf Forensics Secur 14(6):1419–1431
13. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceeding of CVPR arXiv:1612.08242v1
14. He F, Han Y, Wang H, Ji J, Liu Y, Ma Z (2017) Deep learning architecture for iris recognition based on optimal Gabor filters and deep belief network. J Electron Imaging 26(2):023005
15. Pei Y, Huang Y, Zou Q, Zang H, Zhang X, Wang S (2018) Effects of image degradations to cnn-based image classification. arXiv preprint arXiv:1810.05552
16. Zhao Z, Kumar A (2015) An accurate iris segmentation framework under relaxed imaging constraints using total variation model. In: Proceedings of the IEEE international conference on computer vision (ICCV). IEEE, pp 3828–3836
17. Zhao Z, Kumar A (2019) A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features. Pattern Recogn 93:546–557
18. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
19. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
20. Akilan T, Wu QJ, Zhang H (2018) Effect of fusing features from multiple DCNN architectures in image classification. IET Image Proc 12(7):1102–1110. https://doi.org/10.1049/iet-ipr.2017.0232
21. Poster D, Nasrabadi N, Riggan B (2018) Deep sparse feature selection and fusion for textured contact lens detection. In: Proceeding of international conference of the biometrics special interest group (BIOSIG), Darmstadt, pp 1–5
22. Yadav D, Kohli N, Agarwal A, Vatsa M, Singh R, Noore A (2018) Fusion of handcrafted and deep learning features for large-scale multiple iris presentation attack detection. In: Proceeding of IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), Salt Lake City, UT, pp 685–6857
23. Choudhary M, Tiwari, V, Venkanna, U (2020) Iris anti-spoofing through score-level fusion of handcrafted and data-driven features. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106206
24. Czajka A (2015) Pupil dynamics for iris liveness detection. IEEE Trans Inf Forensics Secur 10(4):726–735
25. Gragnaniello D, Poggi G, Sansone C, Verdoliva L (2015) An investigation of local descriptors for biometric spoofing detection. IEEE Trans Inf Forensics Secur 10(4):849–863
26. Li J, Allinson NM (2008) A comprehensive review of current local features for computer vision. Neurocomputing 71(10–12):1771–1787
27. Daugman J (2003) Demodulation by complex-valued wavelets for stochastic pattern recognition. Int J Wavel Multi-resolut Inform Process 1(1):1–17
28. Nosaka R, Ohkawa Y, Fukui K (2011) Feature extraction based on co-occurrence of adjacent local binary patterns. In: Proceeding of Pacific-Rim symposium on image and video technology. Springer, pp 82–91
29. Tola E, Lepetit V, Fua P (2010) Daisy: an efficient dense descriptor applied to wide-baseline stereo. IEEE Trans Pattern Anal Mach Intell 32(5):815–830
30. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceeding of IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, pp 886–893
31. Gragnaniello D, Poggi G, Sansone C, Verdoliva L (2016) Using iris and sclera for detection and classification of contact lenses. Pattern Recogn Lett 82(2):251–257
32. Hsieh SH, Li Y, Wang W, Tien C (2018) A novel anti-spoofing solution for iris recognition toward cosmetic contact lens attack using spectral ICA analysis. Sensors (Basel) 18(3):1–15
33. Sharifi O, Eskandari M (2018) Cosmetic detection framework for face and iris biometrics. Symmetry 10(4):122-1–122-9
34. Menotti D et al (2015) Deep representations for iris, face, and fingerprint spoofing detection. IEEE Trans Inf Forensics Secur 10(4):864–879
35. Silva P, Luz E, Baeta R, Pedrini H, Falcao AX, Menotti D (2015) An approach to iris contact lens detection based on deep image representations. In: Proceeding of 28th SIBGRAPI conference on graphics, patterns and images, Salvador, pp 157–164
36. He L, Li H, Liu F, Liu N, Sun Z, He Z (2016) Multi-patch convolution neural network for iris liveness detection. In: Proceeding of IEEE 8th international conference on biometrics theory, applications and systems (BTAS), Niagara Falls, NY, pp 1–7
37. Kohli N, Yadav D, Vatsa M, Singh R, Noore A (2016) Detecting medley of iris spoofing attacks using DESIST. In: Proceeding of IEEE 8th international conference on biometrics theory, applications and systems (BTAS), Niagara Falls, NY, pp 1–6
38. Nguyen DT, Pham TD, Lee YW, Park KR (2018) Deep learning-based enhanced presentation attack detection for iris recognition by combining features from local and global regions based on NIR camera sensor. Sensors (Basel) 18(8):2601-1–2601-32
39. Demsar Janez (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
40. Chollet F et al (2015) Keras. https://github.com/fchollet/keras
41. Raghavendra R, Busch C (2015) Robust scheme for iris presentation attack detection using multiscale binarized statistical image features. IEEE Trans Inf Forensics Secur 10(4):703–715
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Vis Pattern Recognit. arXiv:1409.1556v6
43. Czajka A, Moreira D, Bowyer K, Flynn P (2019) Domain-specific human-inspired binarized statistical image features for iris recognition. In: 2019 IEEE winter conference on applications of computer vision (WACV). IEEEpp. 959–967
44. Tan CW, Kumar A (2014) Accurate iris recognition at a distance using stabilized iris encoding and Zernike moments phase features. IEEE Trans Image Process 23(9):3962–3974
45. Yadav D, Kohli N, Doyle JS, Singh R, Vatsa M, Bowyer KW (2014) Unraveling the effect of textured contact lenses on iris recognition. IEEE Trans Inf Forensics Secur 9(5):851–862
46. Doyle J, Bowyer KW (2014) Notre Dame image dataset for contact lens detection in iris recognition. In: Rathgeb C, Busch C (eds) Iris and periocular biometric recognition, Chapter: 12.

Institution of Engineering and Technology (IET), London, pp 265–290

47. Available: https://www.iso.org/committee/313770. Accessed on June 2019
48. Yadav D, Kohli N, Vatsa M, Singh R, Noore A (2019) Detecting textured contact lens in uncontrolled environment using DensePAD. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.