

Article

Not peer-reviewed version

Preprocessing of Iris Images for BSIF-Based Biometric Systems: Binary Detected Edges and Iris Unwrapping

[Arthur Rubio](#) * and [Baptiste Magnier](#)

Posted Date: 18 June 2024

doi: 10.20944/preprints202406.1218.v1

Keywords: Iris Biometric Recognition; Image Preprocessing; Canny Algorithm; Iris Unwrapping; Binarized Statistical Image Features; Hough Transform; Computer Vision



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Preprocessing of Iris Images for BSIF-Based Biometric Systems: Binary Detected Edges and Iris Unwrapping

Arthur Rubio ^{1,†}  and Baptiste Magnier ^{2,3}¹ Department of Computer Science & Artificial Intelligence, IMT Mines Ales, Ales, France; arthurrubio0@gmail.com² EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France³ Service de Médecine Nucléaire, Centre Hospitalier Universitaire de Nîmes, Université de Montpellier, Nîmes, France

* Correspondence: arthurrubio0@gmail.com (R.A.); baptiste.magnier@mines-ales.fr (M.B.)

† These authors contributed equally to this work.

Abstract: This work investigates into the enhancement of iris recognition systems through a two-module approach focusing on low level image preprocessing techniques and advanced feature extraction. The first module is dedicated to the preprocessing of iris images, leveraging the Canny algorithm for edge detection followed by the circle-based Hough transform for precise iris extraction. This process, integral to our methodology, ensures the quality and consistency necessary for the subsequent Binary Statistical Image Features (BSIF) analysis. The second module employs the BSIF technique, incorporating domain-specific filters trained on iris-specific data, for robust biometric identification. By combining these advanced image preprocessing techniques, the proposed method addresses key challenges in iris recognition, such as occlusions, varying pigmentation, and textural diversity. Experimental results on the Human-inspired Domain-specific Binarized Image Features (HDBIF) Dataset composed of 1892 iris images demonstrate the effectiveness of the developed approach, with potential future improvements including adaptive algorithms and machine learning integrations for enhanced performance in diverse and unpredictable real-world applications. The novel contributions of this paper lie in the provision of a reproducible research framework, making the iris-domain-specific BSIF filters, training patches, testing database, and source codes publicly available for further application and study.

Keywords: iris biometric recognition; image preprocessing; canny algorithm; iris unwrapping; binarized statistical image features; hough transform; computer vision

1. Introduction

The field of biometric recognition has rapidly evolved, becoming a cornerstone in both security and personal identification domains. Among various biometric modalities, iris recognition stands out due to its high reliability and uniqueness. Similarly to fingerprints, each individual's iris pattern offers a complex and distinctive structure, making it an ideal biometric marker. In a similar vein, the iconic case of Sharbat Gula [1] –the famous photo of the “Afghan Girl”–, whose iris image became emblematic in biometric identification [7], underscores the significance of iris recognition in the modern world. The motivation behind this study lies in harnessing these unique attributes, using advanced image preprocessing techniques to enhance the accuracy and efficiency of iris recognition systems.

In recent years, the integration of biometric systems into security infrastructures has become increasingly prevalent, necessitating more robust and accurate recognition algorithms. Iris recognition, in particular, has seen widespread adoption due to its non-intrusiveness and high resistance to deception. However, the accurate preprocessing and recognition of iris images pose significant challenges, primarily due to variations in image quality, lighting conditions, and position of the eye in space.

This study aims to address these challenges by introducing an innovative approach that combines two powerful techniques: the Canny edge detection algorithm [5] for precise iris segmentation (unwrapping) and Binary Statistical Image Features (BSIF) [4] for effective iris pattern recognition. By improving the preprocessing steps and employing advanced recognition algorithms, this research seeks to push the boundaries of current iris recognition capabilities, offering enhanced performance in both accuracy and speed.

Furthermore, the development of this methodology also serves an educational purpose, providing insights into practical applications of image processing and machine learning techniques in biometric recognition. The purpose of this work is to contribute to the growing body of knowledge in iris biometrics, paving the way for future innovations and applications in high-security environments.

2. Source Code

The reviewed source code and documentation associated to the proposed algorithms are available from the web page of this article, as well as the on the [Iris-Images-Preprocessing-for-BSIF-Iris-Recognition](#) public GitHub repository. The correspondences between algorithms and source codes are mutually given and compilation with usage instructions are included in the README.md of the repository. The repository also contains the scripts allowing to reproduce the experiments included in various figures.

3. Methodology

3.1. Overview

The proposed methodology is divided into two principal modules, each playing a crucial role in the overall process of iris biometric recognition. The different steps are summarized in Figure 1. These modules are designed to efficiently process iris images, extracting relevant information and comparing them for accurate identification. The 1892 images originate from the WACV-2019 Human-inspired Domain-specific Binarized Image Features (HDBIF) Dataset [9]. More details about the dataset are reported in Table 1.

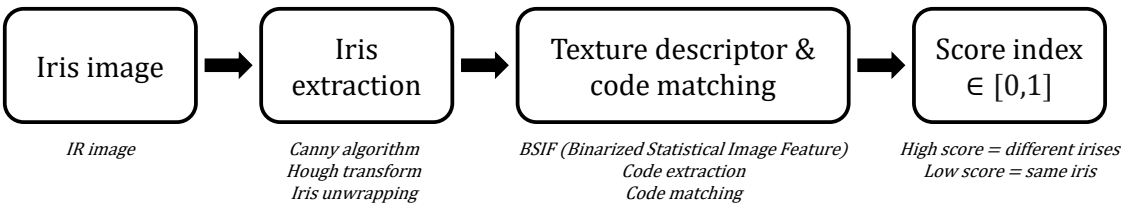


Figure 1. Steps of the process for iris extraction and comparison

Table 1. HDBIF Iris dataset parameters.

Parameters	Value
Number of images	1892
Number of same iris pairs	946
Number of different iris pairs	473
Format	.tiff
Size	640×480
Number of channels	3
Quantification	3×8 = 24 bits

3.2. Module 1: Preprocessing of Iris Images

The first module focuses on the preprocessing of iris images. This process is essential for enhancing the quality and consistency of the images prior to recognition, significantly increasing the accuracy of identification. The preprocessing includes several key steps:

1. **Image Cleaning:** Raw iris images are first cleaned to remove the noise and undesirable artifacts and enhance clarity. This step is crucial for ensuring the accuracy of subsequent stages, and done with a Gaussian and median filter applied to the image, followed by histogram adjustment.
2. **Edge Detection with the Canny Algorithm:** the Canny algorithm [5], a widely-used method for edge detection, is employed to accurately identify the edges of the iris.

3. **Hough transform:** We use the Hough transform [8] to find circles for both the pupil and the iris. We start with the pupil since it is usually clear of eyelash interference. After locating the pupil, we crop the image and introduce a tolerance parameter representing the maximum permissible distance between the centers of the pupil and iris (given their close alignment). Then, we use another Hough transform to detect the iris/sclera boundary. Finally, we extract the iris by applying a binarized filter that captures the area between the two detected circles.
4. **Iris Unwrapping:** This step transforms the iris from its natural annular shape into a rectangular form. This transformation facilitates further analyses, and let the processed image be used for iris recognition using BSIF.
5. **Mask creation:** The unwrapped iris is thresholded to create a custom mask, which will allow the iris comparison module to not compare the part of the iris covered by eyelids or eyelashes.

3.3. Module 2: Iris Recognition

The second module involves iris recognition. After preprocessing, the images are ready for detailed analysis and comparison. This module is based on the work of 4 searchers [2]:

1. **Feature Extraction with BSIF:** Binary Statistical Image Features (BSIF) are used for extracting unique characteristics of each iris. These characteristics serve as a biometric signature. In this case, instead of extracting random textures, we will use textures that have been specifically chosen by a team who manually compared irises to determine if they belonged to the same person.
2. **Comparison and Matching:** Extracted iris codes are compared to determine if they match known samples. This step employs advanced techniques to measure the similarity between images, accounting for possible variations such as eye rotation. Given the number returned (between 0 and 1), you can assess if the two eyes belong to the same person or not.

Together, these two modules form a robust system for iris biometric recognition, offering high precision and reliability in personal identification and security applications. To process and compare irises in our code, we used the HDBIF Dataset. Please follow the instructions [here](#) to get a copy of the test database.

4. Preprocessing of Iris Images

The preprocessing of iris images is a crucial step in the proposed methodology; without it, the method may diverge or yield inconsistent results. It corresponds to the part where we take an image from the HDBIF Dataset and transform it using many algorithms to make it suitable for BSIF iris recognition.

4.1. Dataset Iris Images

The HDBIF Dataset is composed of 1892 iris pictures in .tiff format, coming from 946 different persons. You can observe the various details of the HDBIF dataset in Table 1.

In Figure 2, crucial parameters for iris extraction and unwrapping are observed:

1. The pupil is centered in the frame: we don't have to center the eye by ourselves.
2. Part of the iris is under the eyelid and eyelashes. There is also a white spot. Those parts of the iris should not be analysed in the iris comparison module.
3. Regarding this database, the radius of the pupil can vary a lot (up to 10×) following the light exposition of the eye. This consideration should be taken into account when searching for circles in the image.

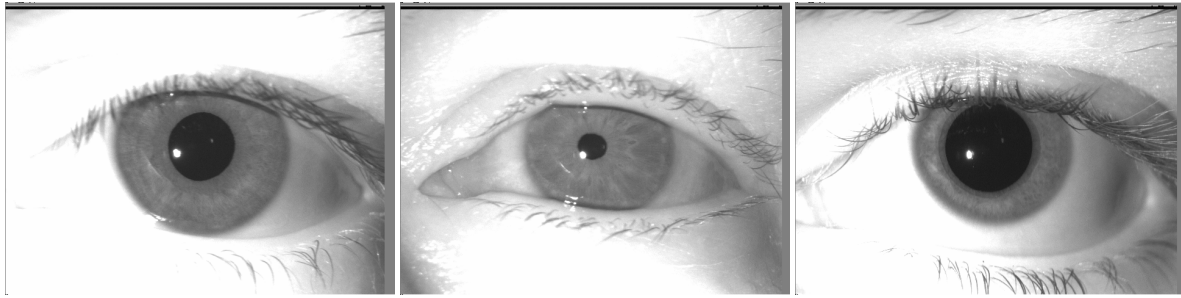


Figure 2. Example of iris images in the HDBIF dataset.

4.2. Iris Parameters Extraction (Radius, Center)

Getting the main parameters of the eye, including its edges, is primordial to extract and unwrap the iris correctly. For that, we will use the Canny algorithm and the Hough transform.

The Canny algorithm [5], a renowned method in edge detection, identifies precise edges within the image, a critical step for accurate feature extraction. This algorithm not only takes into account the gradient magnitude and direction but also facilitates the extraction of edges with basic functions.

After applying the Canny algorithm, the binary thresholding of the edge image is conducted to isolate significant edge features. The resulting binary image undergoes circle detection executed by the Hough Transform. It is worth noting that there are two variations of the Hough Transform, one designed for lines and another for circles [6]. For the purpose of this study, our focus will be exclusively on circle detection. The Hough Transform, tailored to identify circular shapes in images, is pivotal in detecting the circular boundaries of both the iris and pupil, as well as the iris and sclera. The Hough transform method, as implemented in the Matlab native function `imfindcircles`, utilizes specific parameters including sensitivity and the radius range of the circles to detect.

The Canny approach effectively identifies potential boundaries of these anatomical features, considering factors like the partial visibility of the iris due to eyelids or eyelashes, and the typically full visibility of the pupil. Consequently, a higher sensitivity setting is used for detecting the iris circle to account for its possible partial appearance in the image. Detected circles are then visualized on the image with different colors, indicating the boundaries between the iris and pupil, and the iris and sclera. The function also calculates the central coordinates of the iris and determining its inner and outer radii based on the detected circular boundaries.

The gradient magnitude and direction are calculated as usual, as detailed below. Let us consider a 2D isotropic Gaussian filter G_σ of standard deviation σ as in Equation (1):

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad \text{with } \sigma \in \mathbb{R}_+^* \quad \text{and} \quad (x, y) \in \mathbb{R}^2 \quad (1)$$

(x, y) represent the pixel coordinates in the image. First, the smoothed image I_σ is obtained by convolution of the original image I with the Gaussian G_σ . Then, the image derivatives G_x and G_y are computed along the x and y axis in Equation (2.)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I_\sigma, \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * I_\sigma, \quad (2)$$

$*$ represents the convolution product. The gradient magnitude $|\nabla I_\sigma|$ and its direction Θ are computed in Equation (3).

$$\begin{cases} |\nabla I_\sigma| &= \sqrt{G_x^2 + G_y^2}, \\ \Theta &= \arctan\left(\frac{G_y}{G_x}\right). \end{cases} \quad (3)$$

Finally, edges are extracted by computing non maxima suppression in the gradient direction before applying a threshold to obtain binary images, as illustrated in Figure 3b.

In the presented experiments, the Gaussian smoothing in the Canny algorithm takes a standard deviation parameter of $\sigma = 7$.

The Hough transform detects circles by using the equation of a circle with center (a, b) and a radius of R . The equation of a circle is represented in Equation (4).

$$(x - a)^2 + (y - b)^2 = R^2 \quad (4)$$



(a) Original image

(b) Canny algorithm

(c) Hough transform

Figure 3. Iris image after Canny algorithm and Hough transform.

In our approach, the main parameters of the iris are extracted using the Iris parameters extraction Algorithm outlined in Algorithm 1. This algorithm performs a series of image processing steps including image smoothing, histogram normalization, gradient calculation, non-maximum suppression, and Hough transform for pupil and iris detection.

Algorithm 1 Iris parameters extraction**Input:**HDBIF database image img **Output:**Iris inner and outer radius r_{int} ,
 r_{out} Circles center coordinates $(x_{int},$
 $y_{int}), (x_{out}, y_{out})$

- 1: Image smoothing using Gaussian and median filters
- 2: Histogram normalization
- 3: Gradient calculation with Sobel operator
- 4: Apply directional Non-Maximum Suppression (NMS)
- 5: Adaptive thresholding based on local image contrast
- 6: Deleting non-circular components
- 7: Morphological closing ▷ Reinforce circular outlines
- 8: Hough transform for pupil detection
- 9: **if** $pupil_{center} - img_{center} \leq 50$ **then** ▷ Pupil & image center distance tolerance
- 10: **if** $radius_1 \in [20; 80]$ **then**
- 11: $pupil_{radius} \leftarrow radius_1$
- 12: $pupil_{center} \leftarrow center_1$
- 13: **else**
- 14: Increase sensitivity and repeat detection
- 15: **end if**
- 16: **end if**
- 17: Crop the image according to pupil size ▷ Keep iris and avoid error detection
- 18: Hough transform for iris detection
- 19: **if** $iris_{center} - pupil_{center} \leq 20$ **then** ▷ Iris & pupil center distance tolerance
- 20: **if** $radius_2 \in [100; 180]$ **then**
- 21: $iris_{radius} \leftarrow radius_2$
- 22: $iris_{center} \leftarrow center_2$
- 23: **else**
- 24: Increase sensitivity and repeat detection
- 25: **end if**
- 26: **end if**
- 27: **if** max sensitivity reached **and** no iris found **then**
- 28: Define $iris_{radius}$ based on iris size estimation
- 29: **end if**
- 30: Round $iris_{radius}$ and $iris_{center}$ coordinates ▷ For future use

The inner and outer circles are marked on Figure 3c, respectively in blue and red. Many other edges were detected using the Canny algorithm. Those were eyelashes or textures of the iris or skin. The inner and outer circles have the same center: $center = center_{1,2}$

4.3. Iris Extraction

Now that the main parameters of the iris were extracted (radius and center coordinates), we have to extract the iris from the image. The procedure begins with the establishment of a storage folder for processed images. The extraction of essential parameters, including the iris’s external and internal radii and the eye’s center coordinates, are explained in part 4.2.

Following the display of the original image, a cropping operation focuses on the eye region, ensuring the entire iris is included. Then, a filter that isolates the iris ring is created. This filter, designed based on the calculated radii and center, effectively forms a circular mask that targets the iris. The iris is extracted by multiplying the cropped image with this filter, resulting in an isolated and clear representation of the iris. This process, detailed in Algorithm 2 and pictured in Figure 4, not only accentuates the iris for detailed analysis but also prepares it for applications such as biometric identification using BSIF detailed in part 5, where the iris’s unique patterns are crucial for individual identification.

Algorithm 2 Iris extraction

Input:
Image from the HDBIF database I
Iris characteristics (inner and outer radius, center of the circles)
Coordinates of the center of the circles (x_{center}, y_{center})

Output:
File path to the extracted iris image
Cropped squared image around the eye
Filter used to extract the iris
Extracted iris

1: File path creation

2: Crop image

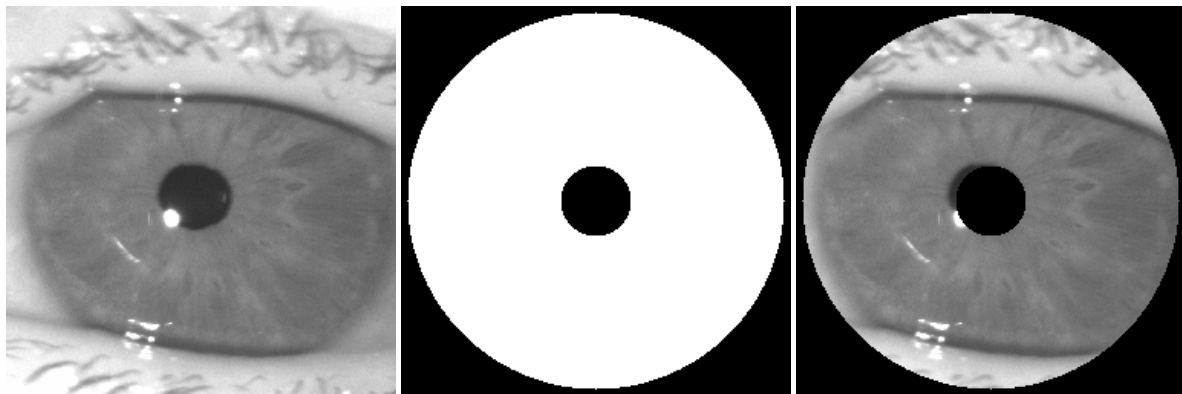
▷ Faster calculations

3: Copped image coordinates: $(x_{min\ max}, y_{min\ max}) = center\ (outer_{radius} + margin)$

4: Filter extraction

5: $filter(i, j) = (r_{inner} \leq \sqrt{(i - x_{center})^2 + (j - y_{center})^2} \leq r_{outer})$

6: $extracted_{iris} \leftarrow I \cdot filter$



(a) Cropped image

(b) Mask used for iris extraction

(c) extractedIris

Figure 4. Extracted iris using Canny algorithm and Hough transform to create the filter.

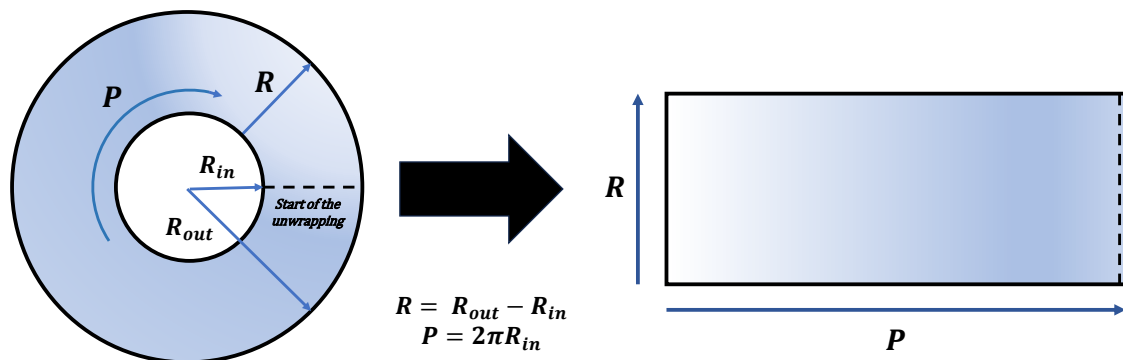
4.4. Iris Unwrapping

The final script in the preprocessing stage is the unwrapping process. This script transforms the circular iris region into a rectangular form, permitting the application of Binary Statistical Image Features (BSIF) [4] in the recognition stage. The unwrapping process ensures that the features of the iris are presented in a consistent and analyzable format. In the second part of the process, we will apply a double threshold to the image to create a binarized masks that will defined the pixels analyzed by the matching code part of the second module.

Initially, the length of a rectangle, intended to encapsulate the unwrapped iris, is determined based on the iris's radius. Coordinate arrays are created to define the spatial points of this rectangle. Central coordinates of the eye are specified, and an angle vector is initialized to assist in generating perimetric points, both inner and outer, around the eye's circumference. These points are then used to create the lines of the rectangle using linear interpolation like showcased in Figure 5, effectively unwrapping the iris into a flat, rectangular image. This image is displayed and resized to a new dimension for compatibility with a specific filter.

Furthermore, a mask is constructed by setting thresholds on the pixel values of the iris image, creating a binary representation that highlights certain features of the iris. This mask is visualized, converted to a logical array for filter compatibility, and then stored as an image file. The combination of geometrical calculations, image preprocessing techniques, and visualization tools allows for a comprehensive approach to iris feature extraction and representation.

Each of these scripts, detailed in Algorithm 3, contributes significantly to the preprocessing stage, ensuring that the iris images are optimally prepared for accurate and efficient recognition.

**Figure 5.** Unwrapping process: from Polar (left) to Rectangular coordinate (right).

Algorithm 3 Iris unwrapping

Input:

Extracted iris

Inner and outer radius of the iris ($innerRadius, outerRadius$)Coordinates of the center of the circles (x_{center}, y_{center})**Output:**Unwrapped iris img Mask of the unwrapped iris $mask$

- 1: Unwrapped iris rectangle:
 $size(rect) = (2\pi \cdot inner_{radius}) \times (outer_{radius} - inner_{radius})$
 - 2: Generation of perimetric points:
 - 3: **for** $i = 1$ to $length(rect)$ **do**
 - 4: $(x_{(1,2)}[i], y_{(1,2)}[i]) = center_{1,2} + innerRadius/outerRadius * \cos(\theta[i])$
 - 5: **end for**
 - 6: Generation of rectangle lines using interpolation:
 - 7: $numPoints = outerRadius - innerRadius$
 - 8: **for** $i = 1$ to $length$ of the rectangle **do**
 - 9: $(x_{line}, y_{line}) \leftarrow linspace((x_1, y_1)[i], (x_2, y_2)[i], numPoints)$
 - 10: $y_{line} \leftarrow linspace(y1[i], y2[i], numPoints)$
 - 11: Linearly interpolate pixel values in positions given by x_{line} and y_{line}
 - 12: Assign interpolated pixel values to i^{th} column in unwrapped iris image img
 - 13: **end for**
 - 14: Double thresholding to obtain iris mask:
 - 15: Calculate *mean* pixel value of the unwrapped iris image.
 - 16: $(lower_{thresh}, higher_{thresh}) \leftarrow (0.5 \times mean, 1.3 \times mean)$
 - 17: Find pixels in unwrapped iris image where $pixel_{value}$ is within threshold range:
 - 18: **if** $lower_{thresh} < pixel_{value} < higher_{thresh}$ **then**
 - 19: $mask_{pixel} \leftarrow 1.$
 - 20: **end if**
-

4.5. Examples of Iris Images: Before and After Preprocessing

To demonstrate the efficacy of our preprocessing techniques, we present examples of iris images before and after the application of our pre-processing scripts. These examples Figure 6 illustrate the significant improvements in image quality and clarity, crucial for accurate iris recognition.

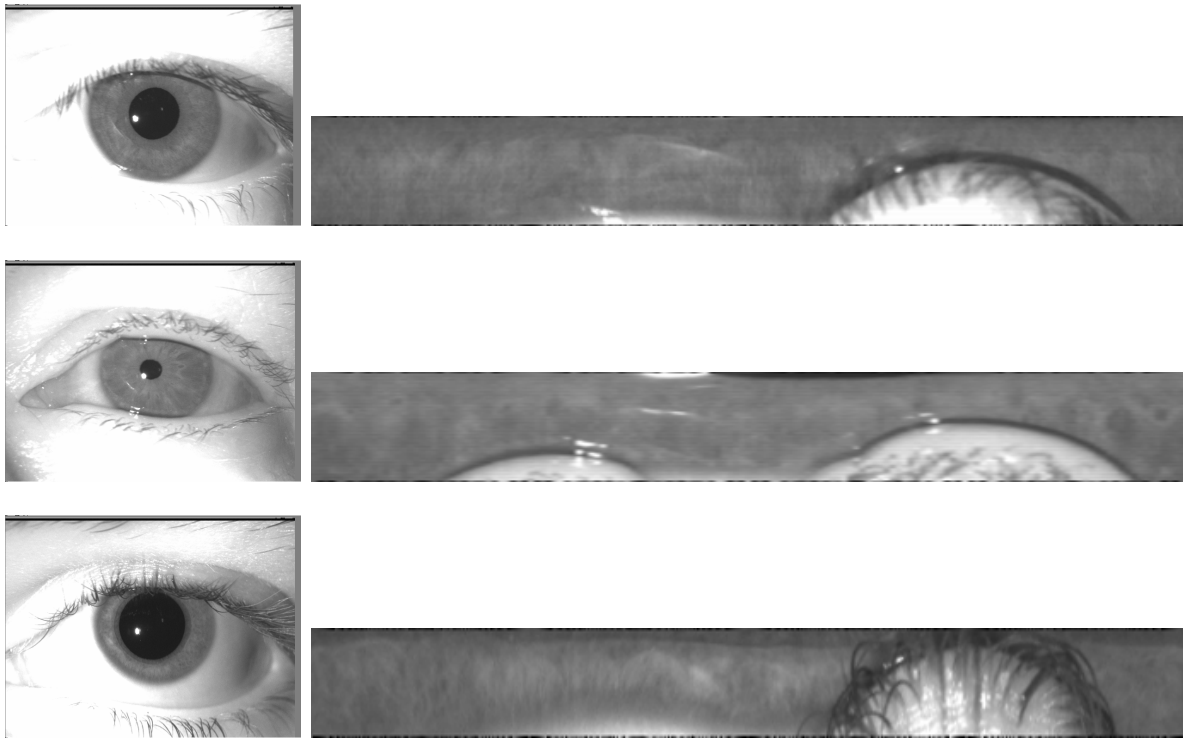


Figure 6. Iris images (left, size 640×480) and their unwrapping (right, size 512×64).

5. Iris Recognition

The iris recognition module heavily relies on the groundbreaking work of researchers from the University of Notre-Dame in the field of iris biometrics [2]. This work plays a crucial role in the development of our module's scripts and algorithms.

A key point of our methodology is the application of Binarized Statistical Image Feature (BSIF) [4] for identity verification through the analysis of pre-processed iris images. The BSIF method employs statistical machine learning to identify key textural patterns in the iris. It computes binary codes for each pixel by projecting local image patches using linear filters and then binarizing these projections. The length of these binary codes is determined by the number of filters used, and the resulting histograms of these codes are instrumental in recognizing textural patterns, thus creating a unique binary representation of these features.

The BSIF filtering starts by getting the filter response with Equation (5):

$$s_i = \mathbf{w}_i^\top \mathbf{x} \quad (5)$$

s_i represents the response of a particular filter. The filter itself is denoted by $\mathbf{w}_i$, and $\mathbf{x}$ is the local image patch under analysis. This equation computes the response of the image patch to a specific filter.

After this, the process continues by combining the different filter responses :

$$\mathbf{s} = \mathbf{W}\mathbf{x} \quad (6)$$

In Equation (6), $\mathbf{s}$ is the collective response vector obtained from applying all filters. $\mathbf{W}$ represents the matrix of all filters used, and $\mathbf{x}$ continues to represent the image patch in the various equations. This equation combines the responses from multiple filters to create a comprehensive response profile for the image patch.

Finally, the following matrix undergoes preprocessing by combining the different filter responses in Equation (7):

$$\mathbf{V} = \mathbf{D}^{-1/2} \mathbf{E}^\top \quad (7)$$

Here, $\mathbf{D}$ and $\mathbf{E}$ are obtained from the eigendecomposition of the covariance matrix, and are integral in constructing $\mathbf{V}$.

This innovative approach, which involved the creation of domain-specific BSIF filters trained on iris-specific data rather than generic natural scenes, has proven more effective than traditional methods. The findings indicate that these domain-specific BSIF features outperform standard ones, showcasing the effectiveness of this tailored approach. Furthermore, they demonstrated the importance of how training data is selected, showing that image patches chosen based on human performance in eye-tracking experiments yield better results than randomly selected patches [2]. This novel method of data selection, guided by human-task performance, is a first in the field and contributes significantly to the advancement of iris recognition technology.

5.1. Extraction of the Binary Code

In the binary code extraction process for iris recognition, a series of steps detailed in Algorithm 4 are followed to transform iris images into a binary format that is essential for identification. Initially, iris images are loaded, including specific masks corresponding to each iris, focusing on fundamental features while filtering out unnecessary parts. The images are then “wrapped” to ensure uniform application of texture filters across all images, a critical step for consistent feature extraction. Next, specific filters tailored for iris texture are selected. These filters are then applied to the iris images, capturing the unique textural patterns of each iris. The output from this filtration is then converted into binary codes, representing the distinctive features of the iris. This binary representation is crucial as it forms the foundation for the subsequent matching and recognition processes in iris biometrics. The code effectively distills the complex patterns of the iris into a format that can be efficiently compared and analyzed, playing a key role in the accurate identification of individuals based on their unique iris patterns.

Algorithm 4 Iris binary code extraction**Input:**Unwrapped iris *img* (see Algorithm 3)

Texture filters

Output:

Unwrapped iris binary code

- 1: Load unwrapped images, masks and iris texture filters
- 2: Number of kernels = numScl ▷ size(texturefilters,3)
- 3: Wrapping of the image:

$r \leftarrow \text{size}(\text{texturefilters}, 1) / 2$ ▷ Size of the filters

$up_{img} = \text{img}(1 : r, :); \quad bt_{img} = \text{img}((end - r + 1) : end, :);$ ▷ Sides

$lf_{img} = \text{img}(:, 1 : r); \quad rt_{img} = \text{img}(:, (end - r + 1) : end);$

$cr_{11} = \text{img}(1 : r, 1 : r); \quad cr_{12} = \text{img}(1 : r, (end - r + 1) : end);$ ▷ Corners

$cr_{21} = \text{img}((end - r + 1) : end, 1 : r); \quad cr_{22} = \text{img}((end - r + 1) : end, (end - r + 1) : end);$
- 4: Loop over the kernels:
- 5: **for** $i = 1$ to numScl **do**
- 6: Select and apply the $numScl_{th}$ filter
- 7: $codeBinary = numScl_{th}filter(wrappedImg) > 0$ ▷ Generate binary code by thresholding the i_{th} $numScl$ filter applied to the wrapped img
- 8: **end for**

5.2. Matching of the Binary Code

The process showcased in Figure 7 and detailed in Algorithm 5 starts by adjusting the binary codes to ensure an effective comparison. This involves removing the margins that were added for the extraction of the binary code.

The system then multiplies both masks of the images to analyze the similarities between irises (as the role of the mask on each picture is to mask regions of the extracted cylinder which are not iris, such as eyelashes or eyelids, as illustrated in Figure 6). After this, for each section of the binary code, the system performs a comparison with a corresponding section from another iris's binary code. This comparison is designed to identify similarities between the two codes. The similarity is quantified based on how many elements of the codes match by computing a score $Score_{bit, shift}$:

$$Score_{bit, shift} = \frac{\sum(\text{XOR}(codeBinary1_{bit, shift}(codeBinary2_{bit})) \text{ AND mask})}{\sum(\text{mask})}. \quad (8)$$

One of the unique challenges in iris recognition is the natural rotation of the eye between different images. To address this, the system includes a mechanism to adjust for eye rotation. This is done by shifting one of the binary codes slightly in different directions and comparing these shifted versions to find the best alignment as showcased in Equation 8.

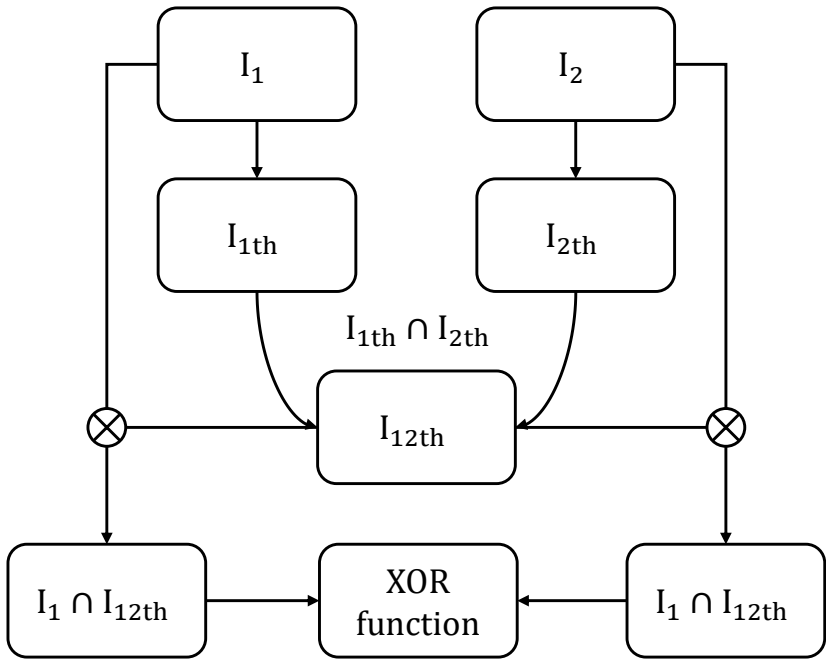


Figure 7. Score comparison process. I_1^{th} and I_2^{th} represent binarized iris images due to thresholding. The intersection ($\cap$) represents element-wise multiplication.

The result of this comparison is a score reflecting the degree of similarity between the two iris images. A lower score (Equation (8)) indicates a higher likelihood that the two images are from the same iris, while a higher score suggests they are from different individuals. This score is crucial in applications like security systems, where it's essential to accurately determine an individual's identity.

To further refine the results, the system calculates comparison scores for different scenarios, such as comparing two images from the same eye (genuine comparison) and comparing images from different eyes (impostor comparison). These scores provide valuable insights into the effectiveness of the iris recognition system and help in fine-tuning its accuracy and reliability.

Algorithm 5 Iris code matching**Input:**

Binary codes of both irises

Masks of both irises

Output:

Iris similarity comparison score

- 1: Remove margins
- 2: Multiply both iris masks: $mask = mask_1 * mask_2$
- 3: Initialize a score matrix $Score_{bit, shift}$
- 4: Initialize the range of possible shifts Max_{shift}
- 5: **for** shift from $-max_{shift}$ to max_{shift} **do**
- 6: **for** each bit in the binary code **do**
- 7: $XOR(codeBinary1_{bit}, shift(codeBinary2_{bit}))$
- 8: Apply masks and use bitwise AND to compare essential parts of the iris codes
- 9: Filtering of resulting XOR values using an AND mask
- 10: $Score_{bit, shift} = \frac{\sum(XOR(codeBinary1_{bit}, shift(codeBinary2_{bit})) AND mask)}{\sum(mask)}$ ▷ The matrix
contains comparison scores for each bit and for each shift
- 11: **end for**
- 12: **end for**
- 13: $Score = \min(\text{mean}(Score))$ ▷ Min avg score for optimal rotation matching across kernels

5.3. Illustrating the Iris Code Comparison Process

A critical aspect of iris recognition is the comparison of iris codes extracted from different images. This process is essential for determining an individual's identity based on their unique iris patterns.

The iris codes, extracted through specific algorithms, are binary representations of the iris textural patterns. The comparison of these binary codes evaluates the degree of similarity by calculating a similarity score based on the number of matching bits between the codes, as shown in Equation (8). A higher similarity score indicates a closer match, suggesting that the two iris images likely belong to the same individual.

One of the challenges in iris recognition is the rotational variance of the iris in different images. To address this, the comparison algorithms include mechanisms to compensate for eye rotation by allowing a degree of shift in the binary code comparison. These techniques together ensure that the iris recognition process is not only accurate but also resilient to common variations in iris images, such as rotation and positional shifts.

6. Results

The efficiency of the proposed methodology was quantified through statistical analysis conducted on the two files referenced in the preceding section. The database was initially subjected to a comprehensive unwrapping procedure. Following this, a meticulous line-by-line analysis was conducted on both files. The BSIF iris comparison algorithm was then applied to the specified pairs of irises mentioned in each respective line. This facilitated the acquisition of comparison scores for each iris pair enumerated within the files. Consequently, distributions of scores for genuine and impostor iris

pairs were established, enabling a rigorous evaluation of the identification accuracy, as illustrated in Figure 8a. You can also see in Figure 8b the distribution of pupil and iris radii sizes and positions.

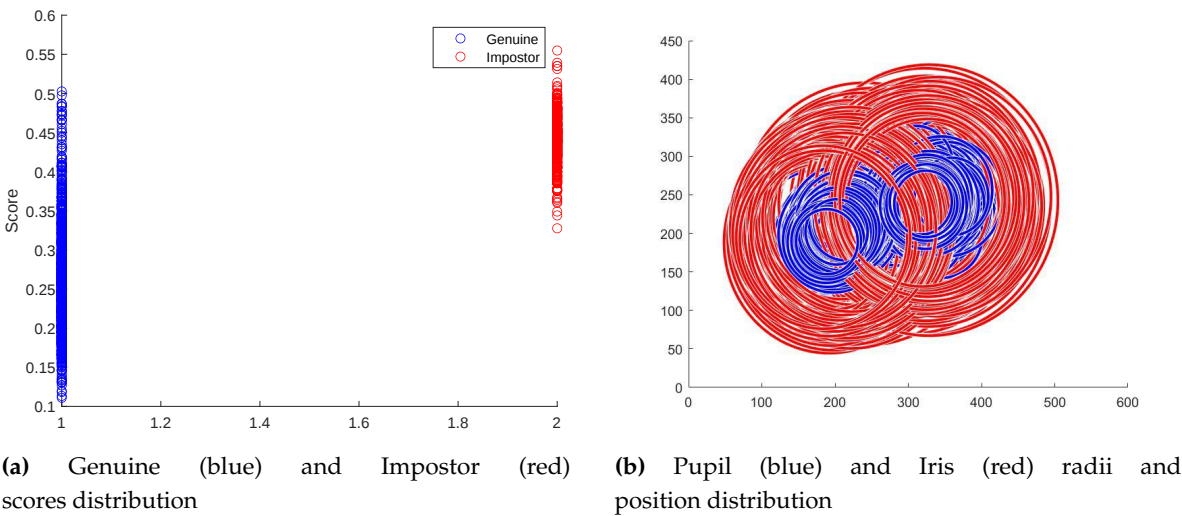


Figure 8. Score distributions and radii/position distributions for iris recognition.

The mean, standard deviation, and variance for each distribution were subsequently computed, culminating in the presentation of the data within the Table 2.

Table 2. Iris distributions metrics.

Distribution	Genuine	Impostor
Mean	0.26482	0.44308
Standard Deviation	0.069778	0.03163

Employing the various metrics delineated in the preceding table, in conjunction with the visual representations of score distributions, a threshold demarcating the distinction between pairs of iris images corresponding to the same individual versus those that do not, was established at 0.385. This threshold enables the calculation of the error percentage attributable to the preprocessing of iris images, which is reported in Table 3.

Table 3. Iris distributions error percentages.

Distribution	Genuine	Impostor
Number of errors	54	16
Number of pairs	946	473
Error %	5.7143%	3.3898%

These results averaging 4,93% global errors validate the robustness and reliability of our proposed method in accurately preprocessing and recognizing iris images, demonstrating its potential for real-world biometric security applications.

7. Discussion

This article has presented a comprehensive study on the preprocessing and recognition of iris images, highlighting significant contributions to the field of biometric security and offering insights into future research directions. The innovative methodologies applied to the preprocessing of iris images have significantly enhanced the quality and uniformity of the data, setting a solid foundation for accurate biometric analysis. Through meticulous cleaning, edge detection, and transformation

techniques, we have successfully addressed the inherent complexities associated with iris recognition, such as occlusions and variations in pigmentation and texture.

The proficiency of the Canny edge detector algorithm combined with the circle Hough transform in delineating the intricate structures of the iris has been demonstrated, proving their efficacy in extracting the vital characteristics required for reliable identification. The unwrapping and masking strategies employed further refine the process, ensuring that the subsequent BSIF analysis operates on data of the highest integrity.

In conclusion, the advancements made in this study, particularly in the preprocessing phase, represent a significant step forward in the field of iris recognition. However, the journey does not conclude here. The continuous evolution of image processing techniques and the integration of machine learning will undoubtedly open new avenues for research, promising even greater accuracy and efficiency in biometric identification systems.

7.1. Limits and Potential Improvements

As seen in Table 3, the iris extraction and unwrapping process is very reliable for BSIF iris comparison. Despite the advancements presented in this work, there remain areas where the preprocessing methodology may be further enhanced. While the preprocessing algorithms have demonstrated efficacy, they are not without constraints. One such limitation is the assumption of ideal conditions within the iris images. Real-world scenarios may introduce variables, such as occlusions or varying lighting conditions, that could impede the accuracy of the feature extraction process. Additionally, the present algorithms may not adequately account for the high variability in iris pigmentation and texture across different populations. Future iterations of this methodology could benefit from machine learning techniques paired with a broader dataset that includes a wide range of iris conditions and variations. In summary, while the presented method constitutes a substantive stride in iris recognition technology, continuous efforts to address the outlined limitations and incorporate the suggested improvements could yield a more versatile and error-resilient system.

Author Contributions: Conceptualization, R.A.; methodology, R.A.; software, R.A.; validation, R.A., and M.B.; writing—original draft preparation, R.A.; writing—review and editing, R.A. and M.B.; visualization, R.A.; supervision, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors declare this project as self-funded.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The HDBIF dataset presented in this research article is available on the [University of Notre-Dame Datasets](#). The code used for the different transformations can be found in the associated [GitHub repository](#). Any additional data, if required, can be obtained by contacting the corresponding author upon request.

Acknowledgments: I would like to thank Lucas Rivière, another student of IMT Mines Alès, for his precious help during the premises of this work.

Conflicts of Interest: The authors declared no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BSIF	Binarized Statistical Image Features
HDBIF	Human-inspired Domain-specific Binarized Image Features

References

1. Strochlic, N. Famed 'Afghan Girl' Finally Gets a Home. *National Geographic* **2017**.
2. Czajka A.; Moreira D.; Bowyer K.W.; Flynn P. Domain-Specific Human-Inspired Binarized Statistical Image Features for Iris Recognition. *IEEE Winter Conference on Applications of Computer Vision (WACV)*; **2019**; Waikoloa, HI, USA; pp. 959–967.

3. Tapia J.E.; Perez C.A.; Bowyer K.W. Gender Classification From the Same Iris Code Used for Recognition. *IEEE Transactions on Information Forensics and Security*, 11, 2016; pp. 1760–1770.
4. Kannala J.; Rahtu E. BSIF: Binarized statistical image features. *IEEE Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*; 2012; Tsukuba, Japan; pp. 1363–1366.
5. Canny J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*; 6, 1986; pp. 679–698.
6. Duda O.R.; Hart P.E. Use of the Hough transformation to detect lines and curves in pictures. *Association for Computing Machinery*; 15, 1972; pp. 11–15.
7. Daugman J. How these algorithms identified the National Geographic Afghan girl, 18 years later. *Cambridge Computer Laboratory*; 2002.
8. Illingworth J.; Kittler J. A survey of the hough transform. *Computer Vision, Graphics and Image Processing*; 44, 1988; pp. 87–116.
9. University of Notre-Dame Datasets. Available online: [HDBIF Dataset](#) (accessed on 8th June 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.