Iris Image Segmentation (Inner Boundary Detection)

Petraru Denisa-Ionela [1, 2], Stan Ana-Maria[1, 3], Vulcu Angelo [1, 4]

[1] Applied Electronics, UTCN, Strada George Barițiu 26-28, Cluj-Napoca, Romania
[2] Petraru.Va.Denisa@student.utcluj.ro
[3] Stan.Ma.Ana@student.utcluj.ro
[4] Vulcu.Li.Angelo@student.utcluj.ro
Abstract: In humans and most mammals and birds, the iris is a thin, annular structure in the eye, responsible for controlling the diameter and size of the pupil, and thus the amount of light reaching the retina. Iris is found to be a well-protected and age invariant biometric.
Many domains require a reliable identification of people, such as: Security in aviation and finance. Biometric identification methods can identify individuals based either on physiological or behavioral characteristics. Systems which use iris biometric are believed to be very accurate. In this proposed segmentation approach, we approximate the pupil and limbic boundaries with the help of circles.
We first attempt to locate pupil circle using Sobel edge detection on thresholded image and a modified and efficient Hough circular transform.

1. Rezumat în limba română/Summary in an European language

Irisul este structura subțire, inelară a ochiului uman, regăsită de asemenea și la majoritatea mamiferelor, responsabilă pentru controlul diametrului pupilei, și astfel pentru cantitatea de lumină care atinge retina. Irisul reprezintă un element sigur al securității biometrice, fiind neschimbat odată cu vârsta.

În cadrul acestui proiect ne propunem să detectăm bariera internă a irisului, prin diferite procese de segmentare si procesare a imaginilor, utilizând limbajul de programare Python.

Vom trece prin diferite etape de procesare a unei imagini inițiale, pentru a detecta forma circulară a exteriorului pupilei. Dorim să ajungem la o reprezentare cât mai simpla si mai robustă a imaginii, astfel o vom subeșantiona și vom aplica un proces de filtrare ca si prim pas, obținând astfel o imagine binară.

O provocare apare datorită prezenței anumitor pixeli nedoriți, precum cei cauzați de gene, umbre si reflexii. Pentru a rezolva aceasta problemă, aplicăm un operator de umplere cu inundare.

Pentru a detecta cu acuratețe pixelii, este esențial ca aceștia să fie pixeli cu margini puternice. Ca să ajungem la rezultatul dorit, trebuie să aplicăm filtre Sobel, atât pe direcție verticală, cât și orizontală. Imaginile obținute astfel vor fi combinate pentru a crea imaginea magnitudinii gradientului, esențială pentru procesare.

Imaginea finală este obținută după aplicarea unui prag imaginii anterioare, astfel obținând o imagine binară formată doar din pixeli de margine puternică. Asupra acestei imagini se aplică transformarea Hough pentru a identifica marginea internă a irisului.

2. State of the Art

In recent years, iris recognition has gained prominence as a highly suitable and reliable biometric modality, particularly within the private sector. Several iris segmentation algorithms have been introduced by different authors in the literature. Most of these algorithms are based on the circular form of the iris boundaries. The overall recognition performance is affected by the presence of noisy elements such as motion blur, eyelashes, reflections and shadows. In literature, there are described several segmentation techniques that address the said noise artefacts.

These techniques can be classified based on the attributes of iris images into four categories: edge based techniques, histogram and thresholding techniques, clustering techniques and contour evolution methods.

However, the iris recognition system is currently in operation on a global scale, standing out as one of the most advanced segments within biometric recognition technology. Moreover, it proves valuable in overcoming technical challenges in instances where face recognition encounters failures or is unavailable, particularly when a user's face is obscured by masks, a scenario that has become especially relevant in the COVID-19 era.

The iris recognition framework, pioneered by Daugman, serves as the cornerstone for the entire iris recognition technology. A standard iris recognition system typically encompasses the following steps: iris image acquisition, image preprocessing, iris segmentation, feature extraction, and feature matching. Iris segmentation holds a crucial role in achieving a high recognition rate within iris recognition. The precision of iris segmentation, when combined with optimal features and effective recognition schemes, contributes to the refinement of the entire iris recognition system. Conversely, inaccurate iris segmentation, even with superior feature extraction and recognition algorithms, cannot compensate for such shortcomings, resulting in a sharp decline in the performance of the iris recognition system. Consequently, the accuracy of iris segmentation holds enormous significance.

In recent decades, propelled by the rapid advancements in deep learning, numerous studies have explored the application of Convolutional Neural Networks (CNNs) in iris segmentation, iris bounding box identification, and pupil center identification. The latest image segmentation models are often variations of encoder-decoder architectures such as U-Net and fully convolutional networks (FCN).

3. Theoretical Fundamentals

The pupil of the eye is commonly assumed to possess a circular shape and, therefore, is represented as a dark circular region located within the iris. To facilitate the detection and delineation of the pupil, we take an eye image, denoted as "I" and scale it down, followed by applying a thresholding process, resulting in a binary image denoted as "I_t". This binary image effectively isolates the pixels representing the pupil, reducing the search space for locating the pupil boundary (the inner boundary of the iris) to the dark pixels within the image.

However, a significant challenge arises due to the presence of various unwanted pixels, such as those caused by eyelashes, eyebrows, shadows, or specular reflections on the pupil. To address this issue, we employ a morphological flood-filling operator to eliminate any holes present in the binary image "I_t". This operation effectively removes specular reflections within the pupil region and other dark artifacts caused by eyelashes, resulting in the improved binary image denoted as "I_{tf}".

To accurately identify the pixels at the boundary of the pupil, it is essential that they correspond to strong edge pixels. To achieve this, we employ Sobel filters in both the horizontal and vertical directions on the binary image "I_{tf}". This process yields gradient images, denoted as "I_{gh}" and "I_{gv}", respectively. Subsequently, these gradient images are combined to create a gradient magnitude image denoted as "I_g", which is essential for further analysis and processing.

$$I_g(x,y) = (I_{gh}^2(x,y) + I_{gv}^2(x,y))^{\frac{1}{2}} \qquad (1)$$

In the final step of the process, we apply a threshold to the gradient magnitude image "I_g" based on the gradient magnitude, resulting in a binary image denoted as "I_{gb}". This binary image exclusively contains strong edge pixels, providing a focused representation of the prominent edges within the eye image. To determine the orientation of an edge at any specific point (x, y) within the binary image "I_{gb}", we utilize the gradient images "I_{gh}" and "I_{gv}". These gradient images are employed to calculate the direction or orientation of the edges present in the "I_{gb}" image.

$$\Theta\ (x,y) = tan^{-1}(\frac{I_{gv}(x,y)}{I_{gh}(x,y)}) \qquad (2)$$

The conventional Hough transform has been enhanced and applied to identify the boundary of the pupil exclusively using the strong edge pixels, which are depicted as "White" in the binary image "I_{gb}". This improvement substantially lowers the time complexity of the Hough transform by diminishing the search area. Every "White" pixel in "I_{gb}" is assumed to be on some circle; hence there are two potential pupil center points (c_1^x, c_1^y), and (c_2^x, c_2^y), on opposite sides of tangent to the circle at that point.

These center points are calculated as follows:

$$c_1^x = x + r * \sin\left(\Theta\ (\mathrm{x,y}) - \frac{\pi}{2}\right) \qquad (3)$$

$$c_1^y = y + r * \cos\left(\Theta\ (\mathrm{x,y}) - \frac{\pi}{2}\right) \qquad (4)$$

$$c_2^x = x - r * \sin\left(\Theta\ (\mathrm{x,y}) - \frac{\pi}{2}\right) \qquad (5)$$

$$c_2^y = y + r * \cos\left(\theta\ (x,y) - \frac{\pi}{2}\right) \qquad (6)$$

A straightforward voting strategy is employed for the 3-tuple, which consists of the pupil center's ordinate, abscissa, and radius. To facilitate this, a 3-D array denoted as "A" is created to store the votes for all possible tuples, where A(c^x, c^y, r) signifies the count of edge pixels situated on the circle defined by the pupil center (c^x, c^y) with a radius of r. The parameters that yield the maximum number of votes determine the coordinates of the center and the radius of the inner boundary of the iris. The inner boundary localization algorithm is detailed in Algorithm 1.

In the process, appropriate thresholding is applied to the original image "I" to generate the binary image "I_t." This binary image is further processed to eliminate specular reflections, resulting in the image "I_{tf}." Sobel edge detection is then performed on "I_{tf}" to obtain the gradient images "I_g", "I_{gh}", "I_{gv}" and the binary edge image "I_{gb}," which exclusively contains strong edges.

Potential center points (c_1^x, c_1^y) and (c_2^x, c_2^y) for the "White" points (x, y) in "I_{gb}" are calculated using equations (3)-(6). Votes are incremented and recorded in the accumulator array "A" for these potential centers. Ultimately, the algorithm produces the essential parameters: (c_p^x, c_p^y), and p_r.

The Hough transform is computed by drawing circles of a given radius at every point in the edge image. For every point where the perimeter of a drawn circle passes, the coordinate was incremented by 1. This was done for every circle drawn to create an accumulation array. A circle is indicated by peaks in the accumulation array (Hough space). Detection of circles using this transformation requires knowledge of the radius .As we don't know the definite radius of the pupil or iris, the transform must be computed for a range of radii. For each radius tested, the location and value of the maximum is stored. The radius with the highest peak indicates the most likely radius and center coordinate for the boundary.

4. Implementation

The first step in achieving the wanted result was making an interface where the user can select exactly the image and the algorithms used, different buttons being created for each instruction. The algorithms are each defined in separate functions, following the steps described in the previous chapter of this report.

We used the Tkinter library for creating a graphical user interface (GUI) application. The application is focused on image processing operations, particularly related to iris segmentation in eye images.

Various other libraries are imported, including Pillow (PIL) for image processing, OpenCV (cv2) for computer vision tasks, NumPy for numerical operations, and Matplotlib for plotting.

We used the following image processing functions:

open_image: Opens an image file using a file dialog and displays it. Converts BMP images to PNG format if needed.

threshold_image: Converts the selected image to grayscale and applies a binary threshold. Displays the original and thresholded images using Matplotlib.

flood_fill_algorithm: Applies a flood fill algorithm to segment an object in the image (e.g., iris). Displays the original, thresholded, and processed images.

edge_detection: Applies Sobel edge detection to highlight edges in the image. Displays the original, thresholded, processed (flood-filled), and Sobel-edged images.

detect_iris: Applies edge detection and Hough Circle Transform to detect and draw circles around the iris in the image. It also displays the result.

The code segment responsible for the graphical user interface (GUI) creation begins by initializing the main Tkinter window, denoted as root, where the application's title and dimensions are set. Additionally, a background image is loaded to enhance the visual appeal of the GUI. The interface features several buttons, each assigned to distinct image processing functions such as image selection, thresholding, flood fill, Sobel edge detection, and application exit. Event handlers are implemented to respond to user interactions with these buttons, triggering the corresponding image processing functions. To enhance user experience and organization, the GUI incorporates labels and buttons arranged in a structured layout, ensuring clarity and ease of navigation. Finally, the main Tkinter event loop is initiated, allowing the GUI to actively run and dynamically respond to user inputs, facilitating a seamless and interactive experience.

Advantages:

High accuracy in Iris recognition: The Hough transform can accurately locate the iris boundary, which is crucial for subsequent steps in iris recognition.

Iris recognition systems have various applications in the medical field, especially in areas requiring precise identification and professional image extraction.

Iris recognition technology offers a highly secure and reliable method for patient identification in healthcare facilities. By capturing and storing unique iris patterns, these systems ensure accurate patient identification, reducing errors in record-keeping, preventing identity theft, and enhancing the overall safety of medical procedures.

Disadvantages:

Our algorithm cannot process every single image correctly because the eyelashes present a problem in performing the Flood-Filling.

The Hough transform presents a high sensitivity to parameters, such as threshold values for circle detection and accumulator resolution.

The Hough transform is specifically designed to detect circular shapes or their variations, which might not fully encompass the complex and irregular shapes sometimes found in the iris.

5. Experimental Results

When running the program, the graphical user interface opens, showing the chosen background image and the buttons needed for choosing the image that will be modified according to the instructions, and for all the algorithms.

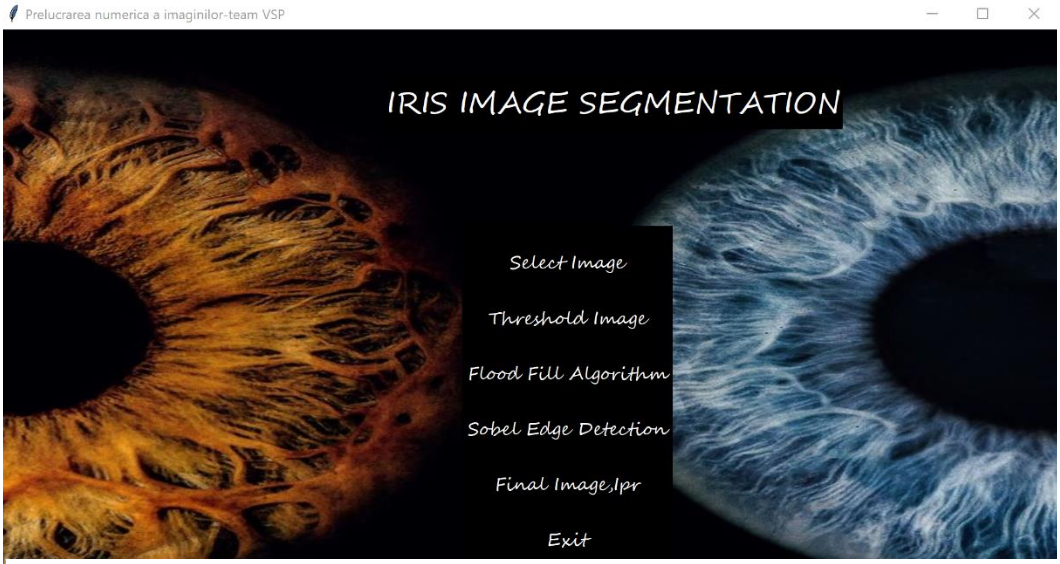

Figure 1 Graphical user interface for the algorithm

When pressing the “Select image” button, the folder with the chosen images opens, making it easy to select the wanted item.

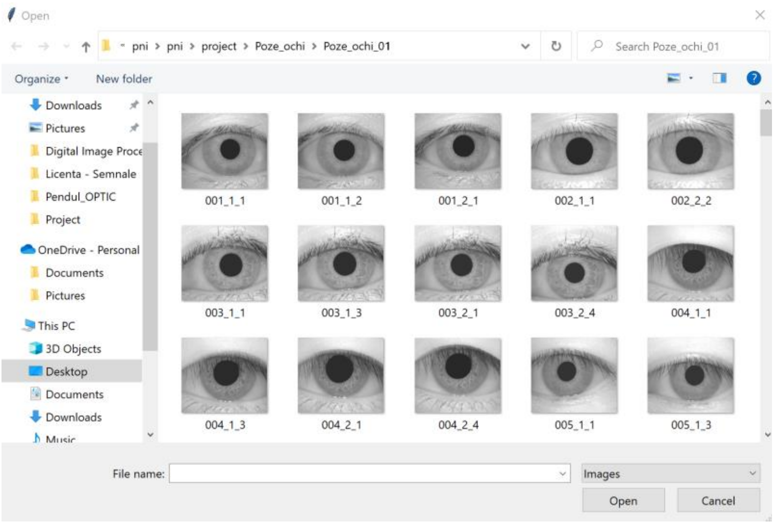

Figure 2 Selection of images for performing the algorithm

The following buttons implement the algorithms used step by step, and the button “Sobel Edge detection” displays all the steps of the image processing. The “Final Image,Ipr” displays the final result over the initial image.

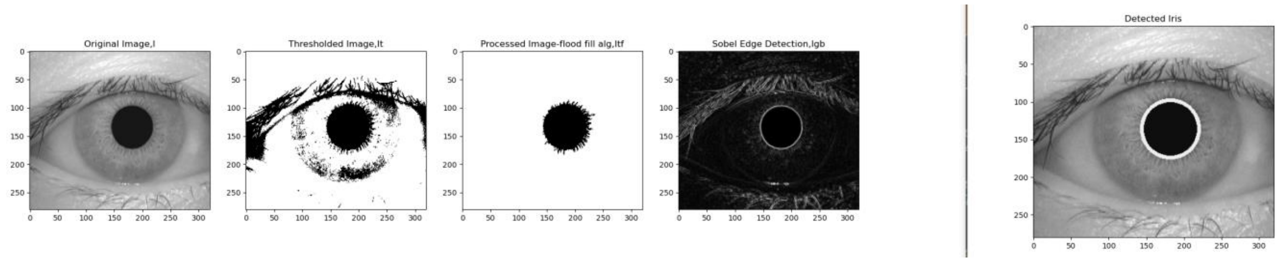

Figure 3 All the transforms that the image goes through: Original image, Thresholded image, Flood-filled image, Sobel Edge Detection, Final image

6. Conclusions

In conclusion, this Python project leverages the Tkinter library for creating a user-friendly graphical interface that facilitates various image processing operations, particularly focusing on iris segmentation in eye images. By integrating the capabilities of Pillow, OpenCV, NumPy, and Matplotlib, the application enables users to open images, apply thresholding, perform flood fill algorithms, and conduct Sobel edge detection for enhanced visualizations. The interface is thoughtfully designed with strategically placed buttons and labels to streamline user interaction. Additionally, the incorporation of background images contributes to the aesthetic appeal of the application. The implementation showcases the synergy between different image processing techniques, providing a comprehensive tool for users interested in exploring and understanding digital image analysis. Even if it has its flaws, overall, this project successfully combines functionality and user interface design to create a versatile and accessible platform for image processing purposes.

7. References

1. Fast Algorithm and Application of Hough Transform in Iris Segmentation, School of Automatic Control, Northwestern Polytechnic University, Xi'an 710072, CHINA

Fast algorithm and application of Hough transform in iris segmentation | IEEE Conference Publication | IEEE Xplore

2. A Fast Iris Localization using Inversion Transform and Restricted Circular Hough Transform, Saiyed Umer Electronics and Communication Sciences Unit Indian Statistical Institute, India

A fast iris localization using inversion transform and restricted circular Hough transform | IEEE Conference Publication | IEEE Xplore

3. Multispectral Iris Recognition Utilizing Hough Transform and Modified LBP, Khary Popplewell, Kaushik Roy, Foysal Ahmad, and Joseph Shelton Center for Advanced Studies in Identity Science North Carolina A&T State University

Multispectral iris recognition utilizing hough transform and modified LBP | IEEE Conference Publication | IEEE Xplore

4. Iris Segmentation and Recognition Using Circular Hough Transform and Wavelet Features Caroline Houston, Rochester Institute of Technology

(2010) Caroline Houston - Iris Recognition.pdf (rit.edu)

5. Iris Recognition using Hough Transform and Neural Architecture Search Network, Ashutosh Soni, Tushar Patidar, Rajesh Kumar M, Bharath K P, Balaji S, Rajkamal Rajendran, 2021 Innovations in Power and Advanced Computing Technologies (i-PACT)

Iris Recognition using Hough Transform and Neural Architecture Search Network | IEEE Conference Publication | IEEE Xplore