Shapley Value-Driven Optimization of Gabor Filters for Iris Recognition

Thimo Veldema

Submisstion date: Oktober 8, 2025

Iris recognition is a highly reliable biometric method due to the stability and uniqueness of iris patterns. John Daugman's iris code[1], based on Gabor filter responses, remains the foundation for most modern systems. However, the lack of transparency regarding the original filters poses challenges for optimization. This work introduces a Shapley[2] value-based approach to evaluate and refine filter selection. By quantifying each filter's contribution to recognition accuracy, redundant filters are eliminated, resulting in a more compact and effective filter set. The study also investigates the impact of rotational misalignments and vertical offsets, which commonly arise from imperfect segmentation. Experiments conducted on multiple datasets, including CASIA-IrisV1 and IrisV3, demonstrate that the proposed method achieves good performance, reaching an equal error rate of 0.05% and 0.67% respectively. Results highlight the critical role of rotation compensation, while vertical offsets show minimal influence.
Iris Recognition; Daugman; Shapley Values; Gabor Filters;

1. Introduction

Iris recognition has emerged as one of the most reliable and non-invasive biometric identification methods. The intricate patterns of the iris remain largely stable throughout a person's lifetime, making it a highly distinctive and enduring biometric identifier. These properties have driven widespread adoption of iris recognition systems in border control, and personal authentication applications.

In the early 1990s, John Daugman introduced the concept of the iris code, a compact 2048-bit representation of iris features that enables rapid and accurate matching. The encoding process relies on complex Gabor filters to extract texture-based features from the iris image[1]. However, a significant limitation of Daugman's original approach is the lack of publicly available information about the specific filters used.

This study explores a strategy to optimize these filters using Shapley value-based feature importance, a technique derived from cooperative game theory[2]. Shapley values provide a way to evaluate the contribution of each filter to the overall performance of the recognition system, enabling a more data-driven approach to filter tuning.

In addition to investigating the effectiveness of Shapley values for filter optimization, the rotational misalignments and vertical offsets impact on iris recognition accuracy is examined. These variations often occur due to imperfect iris segmentation and can significantly degrade system performance.

The research presented in this paper seeks to answer two key questions:

- Can Shapley value-based feature importance be used to effectively tune the Gabor filters in Daugman's iris recognition algorithm?
- To what extent do rotational and vertical offsets affect the performance of iris recognition?

2. Background

2.1. Iris Code

Daugman's iris recognition algorithm operates by first segmenting the iris and transforming it into a normalized, dimensionally consistent format using a rubber sheet model. To handle occlusions, the upper and lower eyelids are detected and masked. Feature extraction is then performed by applying a set of 2D Gabor filters of varying frequencies, orientations, and sizes. These filters are multiplied to the normalized iris image, and the phase information of the complex response is used to generate a binary iris code. Finally, iris codes are compared using the Hamming distance to determine desimilarity[1].

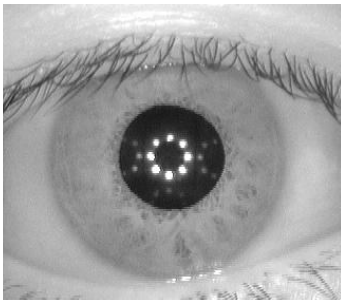

Figure 1.; Infrared closeup of an Eye[3]

Segmentation in Daugman's method is performed using the integro-differential operator, which acts as a circular edge detector. It searches over the image domain for circular paths with maximal change in pixel intensity, effectively locating the iris boundaries.

Eyelid boundaries are also detected using the integro-differential operator. However, rather than fitting circular contours, the operator is adapted to detect parametric spline curves, with parameters estimated via statistical methods to model the upper and lower eyelids[1].

Figure 2.; Iris band from figure 1[3]

After successful segmentation, the iris region is normalized using the rubber sheet model (Figure 2). In this transformation, the angular dimension is mapped along the horizontal axis, while the radial distance from the pupil boundary is mapped along the vertical axis. This normalization compensates for variations in pupil dilation and imaging distance.

To generate the iris code, the normalized image is multiplied pointwise with a family of complex 2D Gabor filters. Each filter responds to specific spatial frequencies and orientations. The result of the multiplication is a complex number for each patch, and only the phase of the response is used for encoding. The phase is quantized into two bits: one for the real part and one for the imaginary part, based on whether each component is greater than or equal to zero. This encoding ensures robustness, where a one-bit difference corresponds to a phase shift of 90°, and two-bit difference corresponds to a 180° shift[1].

Given two iris codes, identity is determined by computing the Hamming distance, which measures the normalized bit difference between two binary codes. To account for occlusions (e.g., eyelids), a mask is applied to exclude unreliable bits from the comparison:

$$HD = \frac{||(C_1 \oplus C_2) \wedge M_1 \wedge M_2||}{||M_1 \wedge M_2||} \tag{1}$$

Where $C1$ and $C2$ are the binary iris codes, and $M1$, $M2$ are their corresponding masks indicating valid (unoccluded) bits.

2.2. Shapley Values

Shapley values originate from game theory and are used to fairly distribute gains or costs among players in a cooperative game, based on their individual contributions[4].

The key idea is to evaluate each player's impact by considering all possible combinations of players. Let's walk through a simple example with two players: **A** and **B**.

- Player A plays alone and scores 70.
- Players A and B play together and score 90.
- At first glance, it seems that B contributes only $90 - 70 = 20$.
- However, if Player B plays alone and scores 100, this changes the interpretation.

In this case, B performs better alone than when paired with A. This suggests that B may actually be contributing more to the outcome, or even carrying A.

This example shows that it's not enough to consider only the individual performance of each player. The interactions between players play a crucial role. Shapley values account for these interactions by averaging each player's marginal contribution across all possible player combinations. This results in a fairer and more accurate distribution of the total gains or costs.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[v(S \cup \{i\}) - v(S) \right]$$

- N is the set of all players.
- S is a subset of players not including player i.
- $v(S)$ is the value (or payoff) of the coalition S.
- $v(S \cup \{i\}) - v(S)$ is the marginal contribution of player i to coalition S.
- The term $\frac{|S|!(|N|-|S|-1)!}{|N|!}$ is a weighting factor based on how many different orders the coalition S can appear in.

For each permutation of the players, we consider the set of players that come before player i, compute the value of the coalition with and without i, and take the difference (i.e., the marginal contribution of i). This contribution is then weighted by a factor based on the number of such permutations. The Shapley value of player i is the sum of these weighted contributions over all possible permutations.

3. Methodology

3.1. Implementation

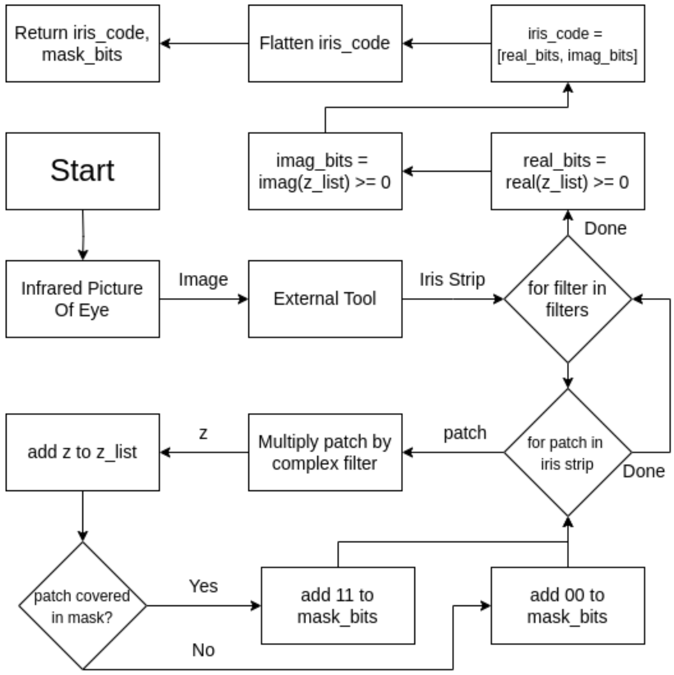

Figure 3.; Iris Recognition Diagram

The implementation of Daugman's algorithm is illustrated in Figure 3. The system takes an infrared image of an eye as input. This image is processed using the University of Salzburg Iris Toolkit (USIT) SDK[5], which performs iris segmentation and masking through the Weighted Adaptive Hough and Ellipsopolar Transform (WAHET) method[6]. The output of this stage is a normalized iris band and its corresponding mask, both with dimensions of 512×64 pixels.

The iris band is then divided into smaller patches. Each patch has the same dimensions as the filter used and is positioned based on specified horizontal and vertical strides. These patches may overlap, depending on the stride settings. The stride also determines the number of bits each filter contributes to the final iris code. For instance, a stride of 8×16 results in 256 patches, each contributing 2 bits, yielding a total of 512 bits per filter.

Each patch is element-wise multiplied with a complex-valued filter. The result of this operation is a complex number. This complex value is then encoded into 2 bits: one bit for the sign of the real part ($real(z) \geq 0$), and one bit for the sign of the imaginary part ($imag(z) \geq 0$).

To generate the mask code, each patch is checked against the corresponding region in the iris mask. If the entire patch lies within the valid (unmasked) area, the corresponding mask bits are set to 11; otherwise, they are set to 00.

This process is repeated for every patch across the iris band and for all applied filters. The final output is a binary iris code along with its associated mask code, which can then be used for comparison and recognition purposes.

The code for the implementation can be found at `https://github.com/CatataFish132/iris_recognition.git`[7]

3.2. Filters

In this iris recognition system, 2D complex Gabor filters are employed with some modifications. Typically, Gabor filters use a Gaussian envelope aligned with the orientation of the sinusoidal wave. However, this alignment caused sharp edges near the sides of the square filter patches. To mitigate this issue, the orientation of the Gaussian envelope is fixed rather than rotated, resulting in smoother edge transitions, as seen in the Gabor filter equation 2.

$$
\begin{aligned}
g_1(x, y, \sigma, \gamma) &= \exp\left(-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}\right) \\
g_2(x, y, \lambda, \theta) &= \exp\left(\frac{2\pi i\,(x\cos(\theta) + y\sin(\theta))}{\lambda}\right) \\
g(x, y, \lambda, \theta, \sigma, \gamma) &= g_1(x, y, \sigma, \gamma) \cdot g_2(x, y, \lambda, \theta)
\end{aligned}
\tag{2}
$$

A critical property of these filters is the elimination of DC components[8]. This is achieved by subtracting the mean from each filter, ensuring no particular phase is favored. This enhances the distinctiveness of the resulting iris codes.

Filter selection is guided by Shapley[2] values, which measure each filter's contribution to the overall recognition performance. Filters with higher Shapley values are retained, while those with lower values are discarded.

Each Gabor filter is defined by the following parameters:

- σ: Controls the size of the Gaussian envelope.
- γ: Defines the ellipticity of the Gaussian kernel.
- θ: Specifies the orientation of the sinusoidal wave.
- λ: Sets the wavelength (period) of the sinusoid.
- ψ: The phase offset, which is fixed at 0, as it is not relevant for this application.

Various filters were tested by rotating θ through four angles: 0, $\pi/2$, $\pi/4$, and $-\pi/4$. For each rotation, a range of σ values from 1 to 16 and λ values from 2 to 28 were explored.

Different filter sizes were also evaluated, including 17×17, 33×33, 17×33, 33×17, and 65×17. For each size, the parameters σ, θ, and λ were independently optimized to reduce the complexity of the search space.

Since Shapley values are relative, their calculation requires an initial set of filters for comparison. The process begins with a set of 8 filters: sizes 17×17 and 33×17, each using the four different θ values mentioned earlier.

New filter sizes are then introduced and optimized. Filters with high Shapley values are retained, while those with low values are discarded. Similarly, existing filters with decreasing Shapley contributions are removed. Importantly, the number of bits generated by a filter is also considered, filters producing fewer, more informative bits may be more valuable than larger filters with slightly higher but redundant contributions.

This iterative process results in a compact set of complementary filters. The nature of Shapley values naturally discourages redundancy: filters that add overlapping information tend to have lower values and are thus excluded. The final outcome is a set of diverse and effective filters optimized for iris recognition.

3.3. Rotation

One common reason for mismatches in iris recognition is rotational misalignment. The process of generating an iris code is highly sensitive to rotation errors, which typically occur during the segmentation step. During segmentation, the circular iris is unwrapped into a rectangular "iris band" for processing. However, if the eye is slightly rotated in the original image, this leads to a horizontal shift in the iris band. As a result, the generated iris code differs even for the same eye, leading to a increased desimilarity score during comparison.

To address this issue, the system performs a series of pixel shifts on the iris band, which simulates various rotational alignments of the iris. This process creates a set of rotated iris codes. Each of these codes is compared to the reference code, and the one with the lowest desimilarity score is used. This ensures that even if the original iris image was slightly rotated, the correct orientation can still be found, improving the reliability of matching.

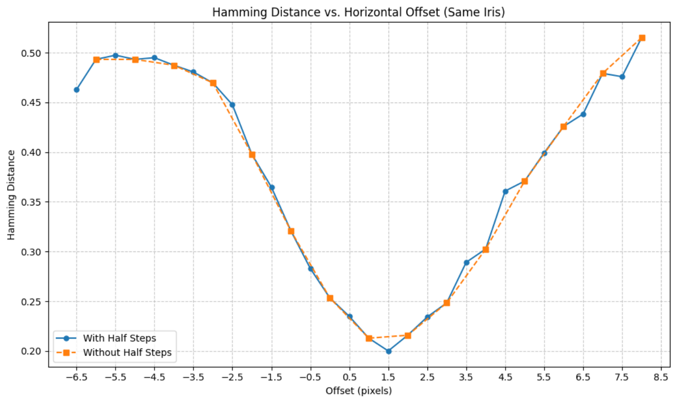

Figure 4.; Hamming distance of the same iris at different rotations

In Figure 4, two images of the same iris are compared. The graph shows a distinct dip in the Hamming distance at a specific rotation, indicating the point where the iris patterns align most accurately. This minimum value corresponds to the best rotational match between the two images.

Interestingly, small deviations from the optimal rotation still produce relatively low Hamming distances. This tolerance is due to the nature of the Gabor filters used in iris encoding, which have a high spatial period. Minor shifts do not significantly affect the filter's phase response, allowing for some flexibility in alignment. However, as the rotational difference increases, the Hamming distance rises to around 0.45, which typically indicates a non-match.

It generally does not matter which iris image is rotated during the comparison—the reference or the query—but slight differences in results can occur. This is because filtering is applied over patches rather than on a strict per-pixel basis, which introduces small inconsistencies depending on which image is rotated. For optimal accuracy, rotating both images would be ideal. However, this approach is computationally inefficient. Therefore, in practice, only the query iris is rotated while the stored reference remains static.

3.4. Vertical Shift

Another factor that can affect the desimilarity score in iris recognition is what can be referred to as a vertical shift. This occurs when the segmentation is inaccurate, not due to incorrect rotation, but due to errors in determining the iris boundaries, particularly the radius or center of the iris.

The inner boundary of the iris (the pupil) is typically segmented accurately, as the contrast between the dark pupil and the iris is quite sharp. However, the outer boundary, between the iris and the sclera (the white of the eye), is often less clearly defined. This makes it more difficult to detect precisely where the iris ends, which can result in a distorted iris band during normalization.

When this happens, the iris band may be stretched or compressed vertically (in the Y-direction), depending on whether the outer radius was overestimated or underestimated. This distortion alters the representation of the iris texture, changing the resulting iris code and thereby increasing the desimilarity score, when the images are from the same eye.

To address this, a similar approach to rotation compensation is used. By trimming the lower or upper rows of the iris band and stretching the remaining rows to fit the original size, we effectively simulate a smaller outer radius or a bigger inner radius.

Since only the normalized iris bands are available at this stage (not the original image), we can't directly expand the outer boundary. Instead, we simulate outward movement by adjusting the comparison iris inward. This has the same effect as moving the iris boundary outwards, however some information is lost in this approach.

As with rotational adjustments, multiple vertically shifted versions of the iris band are compared, and the shift that produces the lowest Hamming distance (i.e., the best match) is selected.

4. Experimental Setup

The CASIA-Iris V1 dataset[9] was primarily used. It consists of 108 unique eyes, with 7 images captured per eye across two separate sessions, resulting in a total of 756 grayscale images, each with a resolution of 320×280 pixels.

The Daugman iris code algorithm was implemented in Python, utilizing the OpenCV library[10] for image processing and NumPy[11] for numerical computations. The complex Gabor filters used in the encoding process were generated based on Equation 2, using the parameter configurations detailed in Table 1.

Size(x,y)	Sigma	Theta	Lambda	Gamma	Stride(x,y)
(17, 17)	5	$\frac{\pi}{2}$	12	1	(8, 16)
(33, 17)	5	$-\frac{\pi}{4}$	20	0.5	(16, 8)
(65, 17)	5	$-\frac{\pi}{4}$	20	0.25	(32, 8)
(33, 17)	5	$\frac{\pi}{4}$	20	0.5	(16, 8)
(65, 17)	5	$\frac{\pi}{4}$	20	0.25	(32, 8)

Table 1.; Gabor Filter Configurations

First, the Wahet method from the USIT SDK[6] is used to extract the iris region from each image. This method outputs an iris band of size 512×64 pixels in grayscale, along with a corresponding binary mask of the same dimensions to indicate occluded or unreliable regions.

From each iris image, an iris code and a mask code are extracted. Each iris code is a 2048-bit binary vector, and each eye is represented in the enrollment database by a single iris code and its corresponding mask.

Next, for each enrolled eye, a set of additional iris codes and masks is generated by applying various rotational shifts and vertical offsets to the normalized images. These transformed representations are stored in two separate 2D arrays, one for iris codes and one for masks, indexed by orientation and vertical shift.

All transformed iris codes are then compared to all entries in the enrollment database, using the masked Hamming distance as defined in Equation 1, which accounts for the binary mask during matching. The resulting desimilarity scores are stored in a 4D array, indexed as (rotation, vertical offset, idx1, idx2), where idx1 and idx2 refer to the indices of the transformed and enrolled iris codes, respectively.

5. Results

5.1. Rotation

Figure 5.; Equal Error Rate (EER) for different rotation ranges

Figure 5 illustrates the relationship between rotation range and the equal error rate (EER). As the rotation range increases, the EER initially decreases sharply and then stabilizes around a range of 21 pixels. Beyond 23 pixels, however, the error rate begins to rise again slightly.

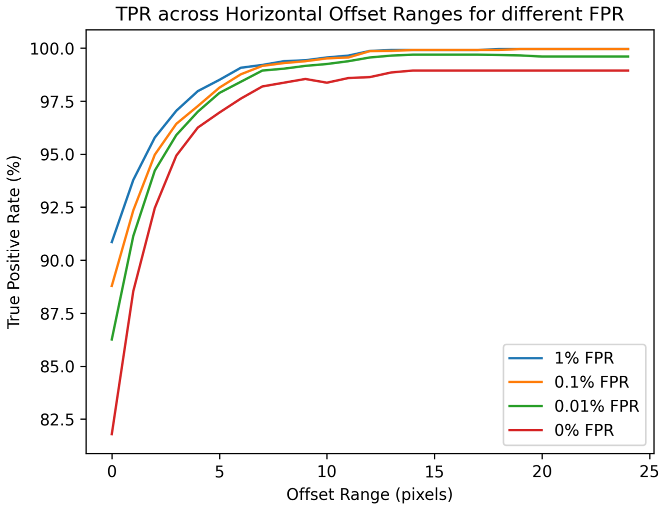

Figure 6.; True Positive Rate (TPR) for various false positive rates across rotation ranges

Figure 6 shows the true positive rate (TPR) for different false positive rates as the rotation range varies. A similar trend is observed: the TPR increases with larger rotation ranges, and the initial gaps between TPR curves narrow as the range increases.

5.2. Vertical Offset

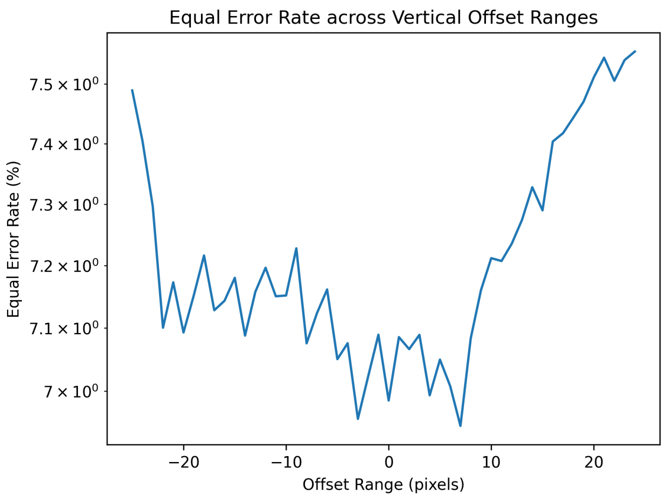

Figure 7.; EER for different vertical offset ranges. Negative values affect the inner iris radius, while positive values affect the outer iris radius.

Figure 7 shows how the EER behaves with varying vertical offsets. Overall, the EER tends to increase as the vertical offset range widens, with occasional isolated decreases that do not form a consistent downward trend.

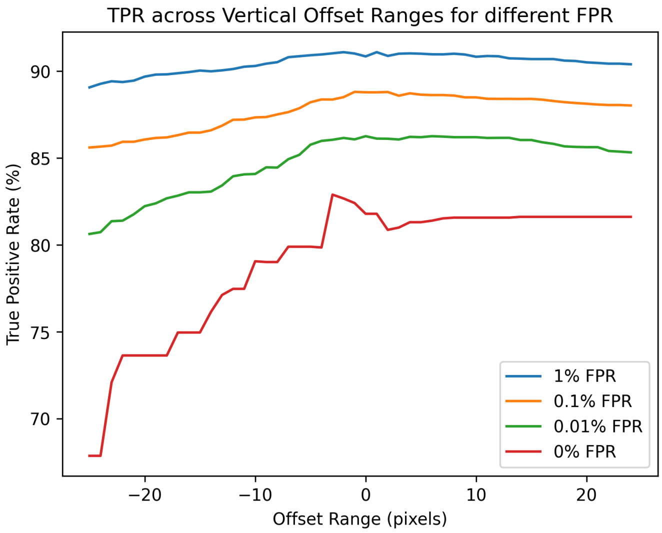

Figure 8.; TPR for different vertical offset ranges. Negative values affect the inner iris radius, while positive values affect the outer iris radius.

Figure 8 presents the TPR across different vertical offset ranges. The TPR generally declines as the range changes, although a slight improvement is observed at the 0 FPR line for a specific range. Notably, altering the inner iris radius tends to cause a more significant drop in TPR.

5.3. System

The results in this section report only the results using rotational offset, as incorporating the vertical offset did not lead to improvements. Consequently, the final system applies only rotation correction.

Dataset	EER	TPR FPR=0	TPR FPR=0.1%	TPR FPR=0.01%
IrisV3	0.0067	0.982	0.993	0.991
CasiaV1	0.0005	0.989	0.9996	0.996
MMU	0.0293	0.890	0.952	0.936

5.3.1 IrisV3

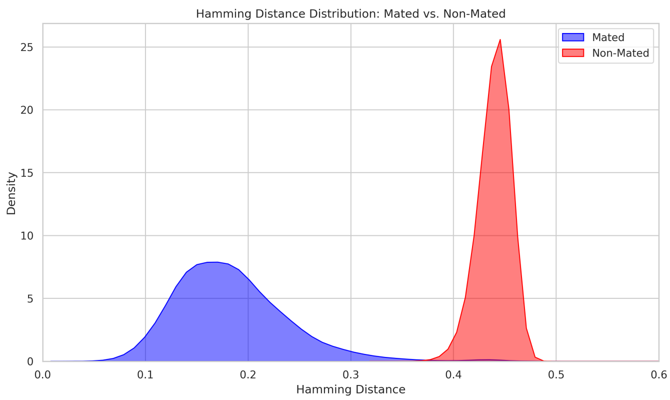

Figure 9.; Hamming Distance Distribution IrisV3

Method	EER (%)	TPR (%) FPR=0.1%	TPR (%) FPR=0.01%
Ours	0.67	99.3	99.1
Deep CNN[12]	0.76	98.8	98.6
QSW[5]	0.99	98.6	98.3
LG[5]	1.220	98.0	96.8
DCT[5]	1.342	97.5	95.8
CG[5]	3.24	92.8	87.5
SIFT[5]	4.161	94.1	93.5
LBP[5]	10.011	71.4	63.0

5.3.2 CasiaV1

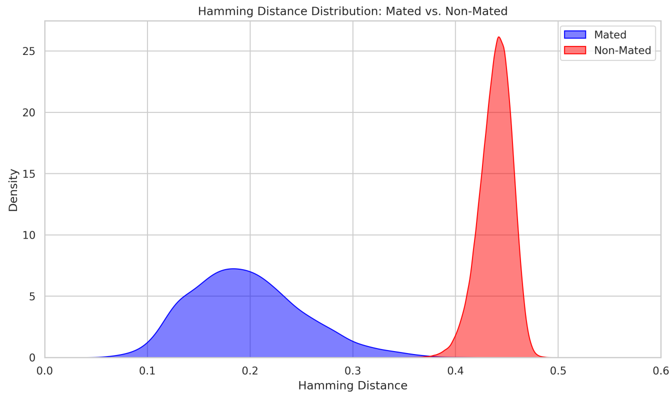

Figure 10.; Hamming Distance Distribution CasiaV1

Method	EER (%)
Ours	0.05
IrisCode[8]	0.32

6. Discussion

6.1. Rotation

The rotation range of the iris plays a crucial role in the effectiveness of iris detection. Expanding this range significantly reduces errors because the matched distribution shifts to the left (indicating lower dissimilarity scores), while the non-matched distribution shifts only slightly left. This increases the separation between the two distributions, making it easier to distinguish matched from non-matched cases.

The error decreases steeply as the rotation range increases, reaching a minimum around 20 pixels, and then starts to rise again beyond 22 pixels. This indicates a narrow optimal range for this dataset. If the range becomes too large, the non-matched dissimilarity scores decrease more than the matched ones, leading to higher error rates. In essence, the rotation adjustment lowers dissimilarity scores for both matched and non-matched cases, but when the decrease is greater for matched cases, the error improves.

Figure 6 also shows that as the rotation range increases, the true positive rate (TPR) rises as well. However, the drop observed after 22 pixels in Figure 5 is not visible here. At smaller rotation ranges, the TPR varies more: for a false positive rate (FPR) of 1%, the TPR is around 91%, but it drops to about 82% when the FPR is 0%. At higher ranges, the TPR stabilizes around 98%. This suggests that matched cases with initially high dissimilarity scores often result from iris rotation misalignment. The small remaining overlap occurs between non-matched cases with unusually low dissimilarity scores and matched cases with unusually high ones.

6.2. Vertical Offset

The vertical offset only reduces the equal error rate (EER) at very specific points, namely at -3 and 7 pixels. This appears to be more of a dataset-specific coincidence rather than a consistent trend. In Figure 8, the TPR generally shows a downward trend in both directions, with the only exception being a slight increase at -3 pixels when the false positive rate (FPR) is 0%. Overall, the vertical offset has little impact on performance and seems highly dependent on the dataset, making it best to disregard in this context.

6.3. System

The system was evaluated using three different datasets: IrisV3, CasiaV1, and MMU. Overall, it performed well across all datasets, though performance on MMU was the lowest. This is most likely due to the lower resolution of its images. CasiaV1 yielded the best results, probably because the system was tuned for that dataset and because it is smaller compared to IrisV3.

Figures 10 and 9 illustrate the distribution of comparison scores. The mean for non-matched pairs is around 0.43. The non-matched distribution is relatively narrow, while the matched distribution is more spread out. Both matched distributions exhibit a tail extending toward the non-matched region, resulting in slight overlap.

6.4. Masking

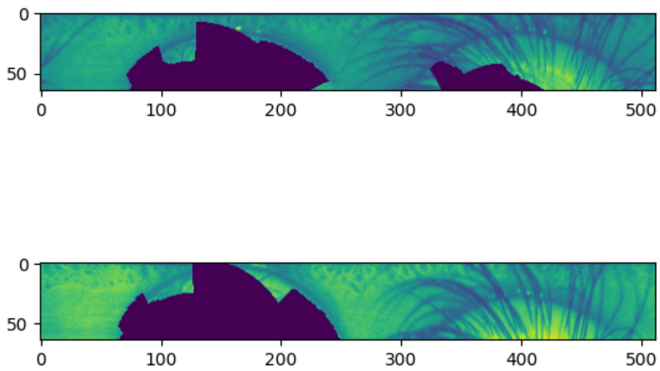

Figure 11.; Iris band with mask generated using WAHET[6]

One of the key factors limiting system performance is the masking process. The eyes are automatically masked using WAHET[6], but this method does not always correctly exclude the eyelids and often fails to mask the eyelashes.

Figure 11 shows one of the worst-performing pairs. Here, the upper eyelids were not masked at all, resulting in a dissimilarity score of 0.404, which is relatively high. Consequently, the system incorrectly classified these two images as belonging to different individuals.

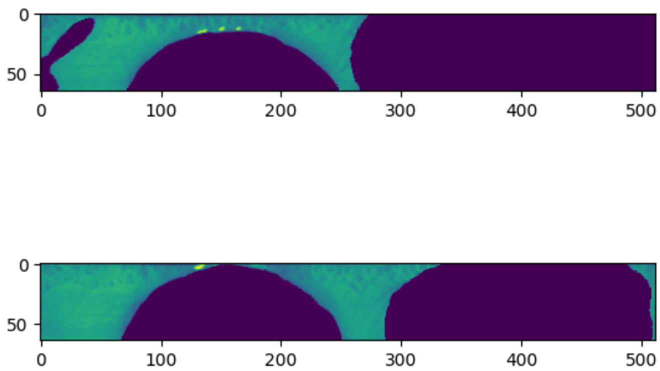

Figure 12.; Iris band with manual made mask including the eyelashes

In contrast, Figure 12 shows the same pair with the mask applied manually, including proper masking of the eyelashes. This adjustment significantly improved the dissimilarity score, reducing it from 0.404 to 0.214.

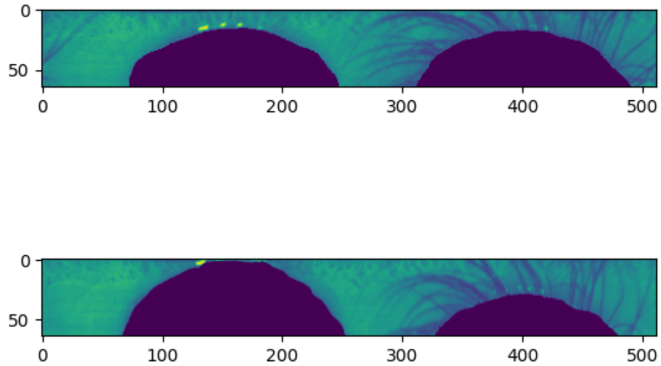

Figure 13.; Iris band with manual made mask excluding the eyelashes

Finally, Figure 13 demonstrates the impact of eyelashes specifically. When the same pair was compared without masking the eyelashes, the dissimilarity score increased from 0.214 to 0.346. This highlights that eyelashes have a substantial influence on the matching performance and should be masked to improve accuracy.

6.5. Improvements

The primary area for improvement is the iris segmentation step. Enhancing the masking process would significantly reduce the dissimilarity scores for matching irises that currently receive relatively high scores, thereby improving overall system performance.

A secondary area for improvement is the rotational alignment during segmentation. At present, the system uses a relatively large rotation range, which lowers the dissimilarity scores for both matched and non-matched pairs. By aligning the iris more accurately using eye landmarks, it may be possible to reduce dissimilarity scores only for matched pairs while keeping non-matched scores unaffected, further enhancing the system's accuracy.

7. Conclusion

This study demonstrates that Shapley value based feature importance provides an effective method for optimizing Gabor filters in Daugman's iris recognition algorithm. By identifying and retaining the most informative filters, the system achieved good performance across multiple datasets. The method reached an equal error rate (EER) of 0.67% on the IrisV3 dataset, outperforming all the methods in the USIT SDK[5] as well as the Deep CNN approach[12]. On the CasiaV1 dataset[9], it surpassed the Daugman implementation of Kong et al. [8].

The results also show that rotational misalignment is a critical factor influencing recognition accuracy, whereas vertical offsets had only marginal effects, this highlights the importance of robust preprocessing steps to mitigate alignment errors.

8. Future Work

Currently, the filters are applied in a uniform grid pattern across the iris. However, this may not be the most effective approach. The region of the iris closest to the pupil contains the most distinctive details and is also the least affected by eyelids. It would be valuable to investigate the impact of applying filters selectively to these key regions instead of uniformly across the entire iris band. Focusing on the most informative and least masked areas could improve the system's performance.

9. AI

This document was edited and refined with the help of AI tools, such as ChatGPT, for clarity, grammar, and phrasing. All technical content, ideas, and structure were created by me. I have thoroughly reviewed the entire text and take full responsibility for the content.

References

[1] J. Daugman, "How iris recognition works", in *The essential guide to image processing*, Elsevier, 2009, pp. 715–739.

[2] L. S. Shapley, "A value for n-person games", in *Contributions to the Theory of Games, Volume II*, H. W. Kuhn and A. W. Tucker, Eds., Princeton University Press, 1953, pp. 307–317.

[3] Chinese Academy of Sciences – Institute of Automation, *Casia-iris image database v3 (interval, lamp, twins)*, `http://biometrics.idealtest.org/`, Accessed: 2025?08?06, 200?

[4] C. Molnar, *Interpretable Machine Learning Models With SHAP, A Guide With Python Examples And Theory On Shapley Values*, 1st ed. 2023, ISBN: 979-8857734445. [Online]. Available: `https://christophm.github.io/interpretable-ml-book`.

[5] C. Rathgeb, A. Uhl, P. Wild and H. Hofbauer, "Design decisions for an iris recognition sdk", in *Handbook of Iris Recognition*, ser. Advances in Computer Vision and Pattern Recognition, K. Bowyer and M. J. Burge, Eds., second edition, Springer, 2016.

[6] A. Uhl and P. Wild, "Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation", in *2012 5th IAPR international conference on biometrics (ICB)*, IEEE, 2012, pp. 283–290.

[7] T. Veldema, *Iris recognition*, `https://github.com/CatataFish132/iris_recognition`, 2025.

[8] A. W. K. Kong, D. Zhang and M. Kamel, "Introduction to the iriscode theory", en, in *Handbook of Iris Recognition*, ser. Advances in Pattern Recognition, K. W. Bowyer and M. J. Burge, Eds., 2nd ed., Guildford, England: Springer, Aug. 2016, ch. 10, pp. 229–245.

[9] Chinese Academy of Sciences - Institute of Automation, *Casia iris image database v1.0*, `http://biometrics.idealtest.org/`, Accessed: 2025-08-06, 200?

[10] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[11] C. R. Harris, K. J. Millman, S. J. van der Walt *et al.*, "Array programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: `10.1038/s41586-020-2649-2`. [Online]. Available: `https://doi.org/10.1038/s41586-020-2649-2`.

[12] G. Liu, W. Zhou, L. Tian, W. Liu, Y. Liu and H. Xu, "An efficient and accurate iris recognition algorithm based on a novel condensed 2-ch deep convolutional neural network", *Sensors*, vol. 21, no. 11, 2021, ISSN: 1424-8220. DOI: `10.3390/s21113721`. [Online]. Available: `https://www.mdpi.com/1424-8220/21/11/3721`.