Iris recognition: comparison of traditional biometric pipeline vs deep learning-based methods

Lorenzo Spataro **Alessio Pannozzo** **Federico Raponi**

Abstract

Iris recognition has become a cornerstone of biometric identification systems due to its unique patterns, stability, and reliability. This study compares a traditional iris recognition pipeline, implemented using the OpenIris library, with a deep learning-based approach employing a ResNet-50 architecture. The traditional method utilizes Gabor filters for feature extraction and Hamming distance for matching, while the deep learning approach generates feature embeddings and evaluates similarity using cosine metrics. Both approaches were evaluated on the CASIA-Iris-Thousand dataset under identical conditions.

1 Introduction

Biometric recognition systems play a critical role in enhancing security by providing reliable and unique identification methods. Among various biometric traits, the iris has emerged as one of the most promising features for recognition due to its distinctiveness, stability, and robustness. The intricate patterns of the iris, which are unique to each individual and remain largely unchanged throughout a person's lifetime, make it an ideal candidate for biometric recognition systems.

The choice of iris as the biometric trait is motivated by several key factors:

- **Uniqueness**: The texture of the iris contains complex and highly unique patterns, even among identical twins. This makes it exceptionally reliable for distinguishing between individuals.
- **Stability**: Unlike other biometric traits such as fingerprints or facial features, the iris remains relatively unaffected by age, external injuries, or environmental conditions.
- **High Accuracy**: Iris recognition systems have demonstrated extremely low false acceptance and rejection rates, making them suitable for high-security applications.

Potential drawbacks of using the iris as a recognition system include:

- **Sensitivity to Lighting Conditions**: Capturing high-quality images can be difficult in poor lighting environments or when factors such as glasses, contact lenses, or reflections interfere with the view of the iris.
- **Requirement for User Cooperation**: The system relies on users positioning themselves correctly and keeping their eyes open during scanning, which may pose challenges for individuals with disabilities or young children.
- **Risk of Spoofing**: While uncommon, sophisticated spoofing methods such as high-resolution printed images or specially designed contact lenses replicating iris patterns could potentially compromise the system.

1.1 Pipeline

1. **Segmentation**: The first step in the iris recognition system is **segmentation**, which involves isolating the iris region from the captured eye image while excluding surrounding structures such as the sclera, eyelids, eyelashes, and reflections. This is typically achieved using **Daugman's integro-differential operator**(1), a widely adopted method in iris recognition. This technique identifies the circular boundaries of the pupil and iris by maximizing the integral of pixel intensity changes along circular contours. By effectively addressing variations in lighting and occlusions, this approach ensures accurate localization of the iris region, providing a

reliable foundation for the subsequent stages of the recognition process.

2. **Normalization**: To ensure that the extracted features are consistent and comparable across varying imaging conditions, we used the *Rubber Sheet Model* for normalization. This model maps the segmented circular iris region into a fixed, rectangular coordinate system. Each point in the iris is transformed from polar coordinates (r, θ), where r is the radial distance and θ is the angular coordinate, to Cartesian coordinates in a rectangular grid. The mapping ensures that the radial and angular deformations caused by changes in pupil dilation or viewing angle are corrected. This process creates a dimensionally invariant representation of the iris, enabling consistent feature extraction.

3. **Coding (Feature Extraction)**: The next phase involves the extraction of distinctive features from the normalized iris pattern. For this task, we used *Gabor filters*(2), which are highly effective in capturing spatial frequency and orientation information. Gabor filters are applied to the normalized iris image to extract texture information by convolving the image with a series of filter kernels at multiple scales and orientations. This method allows for the capture of detailed texture patterns in the iris, which are essential for distinguishing between individuals. The resulting feature map is then encoded into a binary iris code that uniquely represents the iris texture, enabling efficient comparison and matching in the later stages of the recognition process.

4. **Matching**: In the final phase, the extracted iris code is compared with pre-stored templates in a database to verify or identify an individual. This comparison is performed using *Hamming distance*(3), a widely used metric for comparing binary codes. The Hamming distance measures the number of differing bits between two binary strings. It is calculated by counting the positions where the bits in the two iris codes differ. A lower Hamming distance indicates a closer match between the two iris patterns, while a higher distance signifies a mismatch. The system employs a threshold value to decide whether a match is found or not, minimizing both false acceptance and false rejection rates. The use of Hamming distance provides a simple yet efficient mechanism for comparing the compact binary iris codes generated during feature extraction, ensuring fast and accurate matching.

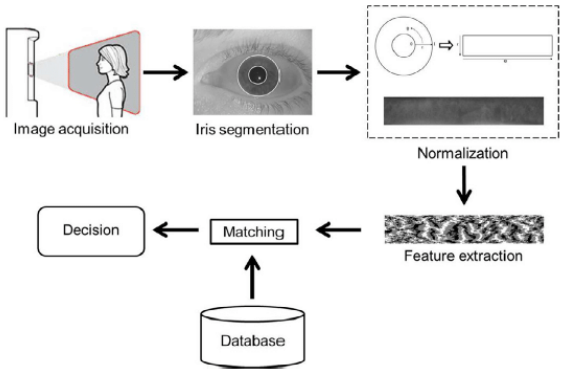

Figure 1: Standard pipeline

1.2 Our Goal

Our primary goal is to systematically compare the performance of a **traditional approach** to iris recognition against a **modern deep learning-based approach**. The traditional pipeline, implemented using the OpenIris library, relies on well-established techniques such as Gabor filters for feature extraction and Hamming distance for matching. In contrast, the deep learning pipeline leverages a **ResNet-50 architecture**(4), which is capable of automatically learning rich and hierarchical features from iris images, with cosine similarity as the matching metric. By evaluating both approaches on the same dataset and under identical experimental conditions, we aim to highlight the strengths and limitations of each method. This comparative analysis will provide valuable insights into the trade-offs between classical, rule-based algorithms and data-driven, neural network-based systems, enabling us to understand their suitability for various real-world scenarios and applications.

2 Project Design

In our project we used the *OpenIris* library for the traditional machine learning approach, and the *ResNet-50 architecture* for the deep learning approach.

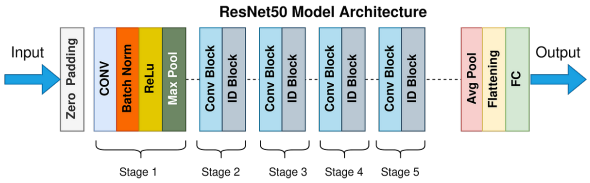

Figure 2: ResNet-50

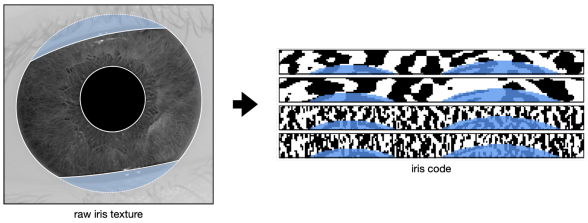

Figure 3: OpenIris Eye Segmentation

2.1 OpenIris Library

To implement the pipeline described above, we utilized the Python library OpenIris (5), an open-source framework specifically designed for iris recognition tasks. OpenIris provides a comprehensive set of tools to process eye images efficiently and accurately, covering all key phases of the recognition system, including segmentation, normalization, feature extraction, and matching. By leveraging OpenIris, we ensured that each phase of the pipeline was implemented with reliable and well-tested algorithms.

2.1.1 Segmentation with OpenIris

The segmentation is first and most critical step in the system. It involves isolating the iris region from the captured eye image by accurately segmenting the key components of the eye while excluding noise and occluding structures such as eyelashes, hair strands, and reflections. For this step, we employed a **novel dual-headed neural network architecture** (6), specifically designed for high-resolution infrared iris images. The architecture consists of two decoders working in parallel:

1. **Geometry Decoder**: Estimates the key geometric elements of the eye, including the pupil, iris, and sclera.

2. **Noise Decoder**: Identifies and processes noise elements that obscure the iris texture, such as eyelashes and reflections.

This dual-decoder design allows for the separate handling of geometry and noise, significantly improving the system's flexibility and efficiency. By decoupling these tasks, the method ensures accurate segmentation even in cases of overlapping or occluding elements, which are common challenges in real-world scenarios. The architecture is based on the DeepLabv3+ framework with a MobileNet v2 backbone, a lightweight yet powerful design optimized for high-resolution image segmentation.

2.2 Feature extraction with a ResNet50

This work presents a deep learning approach for feature extraction in iris recognition, leveraging a ResNet-50 architecture instead of a traditional Convolutional Neural Network (CNN). Unlike traditional feature extraction methods such as Gabor filters, which rely on hand-crafted techniques, ResNet-50 automatically learns hierarchical and discriminative features directly from the input data. This provides a more powerful and flexible representation, capable of capturing complex patterns and variations in iris textures.

During the feature extraction process, the ResNet-50 model processes normalized iris images through its 50 layers, organized into five blocks of convolutional layers, pooling layers, and nonlinear activation functions. These layers progressively extract low-level features, such as edges and textures, and high-level features that capture abstract, unique patterns of the iris. The final output of the network is a **feature vector of length 2048**, where each element is a floating-point value in the range $[-1, 1]$. This vector represents a compact and highly discriminative embedding of the iris pattern, well-suited for efficient matching and comparison in the later stages.

This approach is inspired by the work in *"ThirdEye: Triplet Based Iris Recognition without Normalization"*(7) by Ahmad and Fuller, which demonstrated the effectiveness of ResNet-based architectures for biometric feature extraction.

2.3 Modifications to the ResNet-50 model

The model previously described was pre-trained for a classification task using a softmax loss function. To adapt the ResNet-50 architecture to our specific task, three modifications were required.

The first modification involves replacing the ini-

tial convolutional layer, altering the number of input channels from 3(RGB) to 1. This change is necessary because the images in the dataset used for both training and testing are in grayscale, as they were captured by an infrared sensor. The second modification consists of removing the *softmax* function, as it is not pertinent to the objectives of our task. The final modification is an addition, which introduces output normalization in **L2**[1] form.

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{1}$$

2.4 Tripet Loss Function

In order to ensure that the model generates highly similar embeddings for images belonging to the same class, it was necessary to define a new loss function for the training phase, referred to as *Triple Loss*(8). This function takes three inputs: anchor, positive, and negative. The "anchor" represents the result obtained by passing an image of a user to the model. The "positive" parameter corresponds to the result generated by the model using an image of the same user as the anchor. Finally, the "negative" parameter represents the result obtained from the model using an image of a different user as input. With these three parameters, the loss is computed using **cosine distance**, with a margin of *0.4*. This approach enables the model to more effectively learn how to generate embeddings, thereby improving its ability to distinguish between subjects.

The Triplet Loss with cosine distance can be expressed as:

$$L_{\text{triplet}} = \max\left(0,\ d(\mathbf{a}, \mathbf{p}) - d(\mathbf{a}, \mathbf{n}) + \alpha\right)$$

where:

- $\mathbf{a}$ is the anchor,
- $\mathbf{p}$ is the positive sample (same class as the anchor),
- $\mathbf{n}$ is the negative sample (different class from the anchor),
- $d(\mathbf{x}, \mathbf{y})$ is the cosine distance between two vectors $\mathbf{x}$ and $\mathbf{y}$, defined as:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}$$

- α is the margin ensuring the distance between the anchor and negative is sufficiently larger than the distance between the anchor and positive.

2.5 Training

In order to adapt the ResNet architecture to our dataset, two models were trained, each for 30 epochs. The first model was trained using images that encompassed the entire eye, while the second model was trained on images containing the normalized iris, computed using the OpenIris library. The Adam optimizer was employed for training, with a learning rate set to 1e-3.

2.6 Matching Phase with ResNet-50 Features

For the matching phase, we use the *cosine similarity* as metric. Cosine similarity measures the angular difference between two feature vectors in a high-dimensional space, making it invariant to scale and robust to variations in intensity or contrast.

To facilitate the comparison between the results obtained by the models and those derived using the Hamming distance, modifications were made to the results produced by the cosine similarity measure. The first modification involves inverting the output of the cosine similarity function so that vectors belonging to the same class yield results closer to -1. The second modification involves normalizing the resulting values to the interval [0,1].

3 Dataset

To evaluate the performance of the iris recognition pipeline, we utilized the **CASIA-Iris-Thousand dataset**(9), a widely recognized benchmark dataset in the field of biometric research. This dataset consists of **20,000 high-resolution images of irises** captured under controlled conditions, representing a diverse range of individuals and variations in iris patterns. Our evaluation aims to compare the performance of the **ResNet-50 models** trained on this dataset with the traditional method implemented using the **OpenIris library**. We evaluate two models: one ResNet-50 trained on a dataset of centered eye images and another ResNet-50 trained on a normalized iris dataset. To achieve this, we created three distinct datasets tailored to different stages of the evaluation. The dataset was split into three subsets to ensure a structured and fair evaluation:

- **70% for training**: This portion was used to train the ResNet-50 model, allowing it to learn patterns and features from the iris images.
- **10% for validation**: This subset was used during the training process to monitor the model's performance, tune hyperparameters, and prevent overfitting.
- **20% for testing**: These images were held out during training and used exclusively to evaluate the final performance of the model, ensuring that no data leakage occurred.

The splitting process was performed at the individual level to ensure that no samples from the same individual were present in more than one subset. This approach guarantees that the model's performance is evaluated on unseen identities, providing a realistic measure of its generalization capabilities.

3.1 Segmented Iris Dataset

In the first step, we processed the original images using the OpenIris library to identify and segment the iris region. Of the 20,000 images in the CASIA-Iris-Thousand dataset, 18,725 images were successfully segmented, approximately 93% of the images.

This segmented dataset retains the original circular iris format and serves as input for the subsequent normalisation and feature extraction steps. Images that failed segmentation were excluded from further processing, ensuring a clean and reliable dataset.

3.2 Normalized Iris Dataset

From the segmented dataset, we applied the Rubber Sheet Model to normalise the regions of the iris. This normalisation step transformed the circular regions of the iris into a fixed, rectangular representation by mapping the iris from polar coordinates to Cartesian coordinates. This transformation compensates for variations caused by pupil dilation, changes in viewing angles and other distortions, ensuring that all irises have a consistent format.

The resulting dataset consists of rectangular representations of 18,725 normalised irises, uniform in size and orientation. This dataset is the input for feature extraction, providing a dimensionally invariant representation that allows for the extraction of discriminating features.

3.3 Feature Vector Dataset

In the final stage, we extracted features from the normalized iris dataset using Gabor filters.

These feature maps were then processed to produce feature vectors, compact numerical representations of the iris patterns. Each feature vector contains critical information required for distinguishing between irises and serves as the input for the matching phase. The final feature vector dataset is composed of 18,725 feature vectors, one for each iris, ensuring compatibility with the matching algorithms such as the Hamming distance.

4 Evaluation

Our goal is to clearly define the two primary evaluation tasks:

1. **Verification**: Assess the ability of the system to verify a user's identity based on an iris sample and a claimed identity.
2. **Identification**: Evaluate the system's capability to identify a user from a set of known individuals (closed-set identification) or determine if the user is unknown (open-set identification).

4.1 Test dataset

To evaluate the performance of the deep learning models and the traditional iris recognition pipeline, we utilized 20% of the total dataset, which corresponded to 400 unique identities. Each identity is represented by around 8 samples. These 400 identities were further split into two subsets for the evaluation: 60% were designated as the gallery set, and 40% as the probe set.

4.2 Evaluation metrics

These are the key metrics used for our evaluation

1. **False Acceptance Rate (FAR)**: Percentage of impostor attempts that are falsely accepted. *(Indicates system vulnerability to unauthorized access.)*
2. **False Rejection Rate (FRR)**: Percentage of genuine attempts that are falsely rejected. *(Indicates how often legitimate users are denied access.)*

3. **Equal Error Rate (EER)**: The point where FAR and FRR are equal. A lower EER indicates better performance.

4. **Genuine Rejection Rate (GRR)**: This is the rate of impostors whose identity is correctly rejected (1 - FAR)

5. **Receiver Operating Characteristic (ROC) Curve**: Plots Genuine Acceptance Rate (GAR = 1 - FRR) against FAR. An ideal system has a curve that approaches the top-left corner.

6. **Detection and Identification Rate at Rank-1 (DIR@1)**: Measures the probability of correctly identifying the user at the top of the ranked matches.

4.3 Verification

For the verification task we generated multiple templates per individual in the test dataset and than for each template:

- Perform **genuine verification** by comparing it to other templates of the same individual.
- Perform **impostor verification** by comparing it to templates of all other individuals.

4.3.1 Open Iris Library

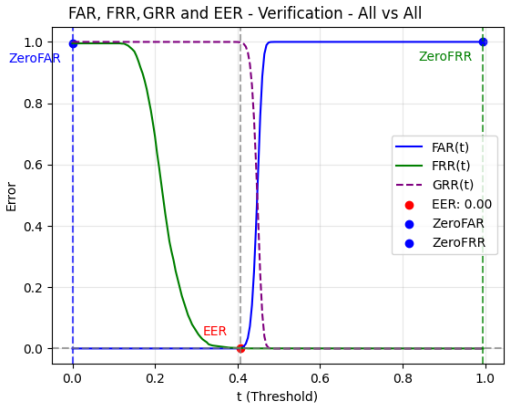

Figure 4: Verification All vs All

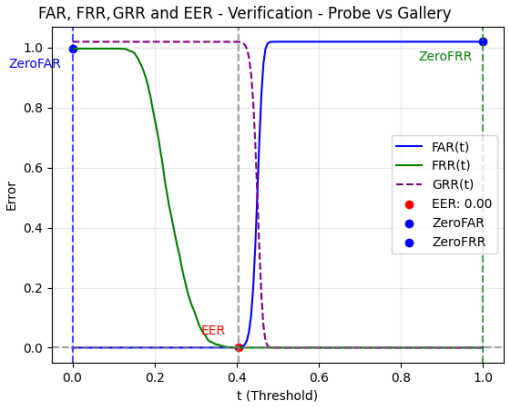

Figure 5: Verification Probe vs Gallery

These graphs illustrate the verification performance across different scenarios, including comparisons between probe and gallery samples as well as broader all-vs-all evaluations. It plots key metrics such as the False Acceptance Rate (FAR), False Rejection Rate (FRR), Genuine Recognition Rate (GRR), and Equal Error Rate (EER) against varying thresholds. The graph highlights critical points like ZeroFAR and ZeroFRR, which indicate the thresholds where FAR and FRR are minimized or reach zero, providing valuable insights into the system's performance.

4.3.2 Resnet-50 with full eye images

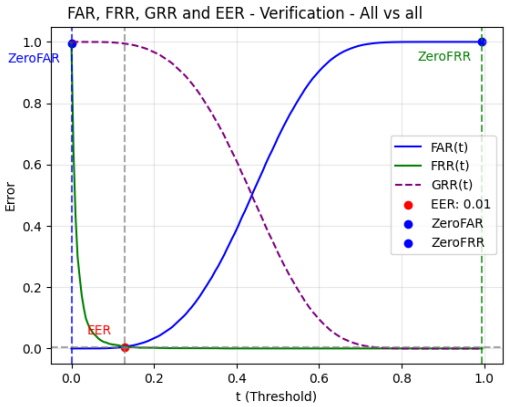

Figure 6: Verification All vs All

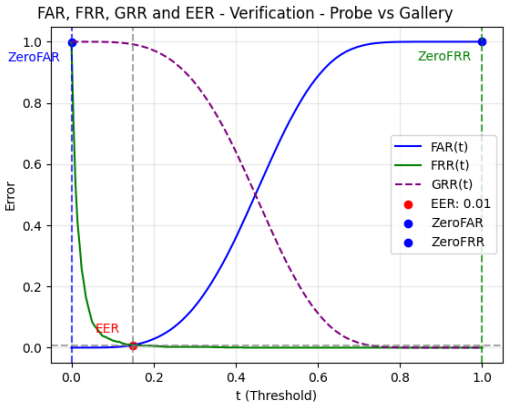

Figure 7: Verification Probe vs Gallery

These graphs present a detailed evaluation of the full-eye ResNet model's verification performance in both "All vs All" and "Probe vs Gallery" scenarios. It plots the False Rejection Rate (FRR) against varying thresholds, highlighting key points where the False Acceptance Rate (FAR) and FRR are minimized to zero. Additionally, the graph marks the Equal Error Rate (EER) at 0.01, showcasing the model's high accuracy and effectiveness in these scenarios.

4.3.3 Resnet-50 with normalized eye images

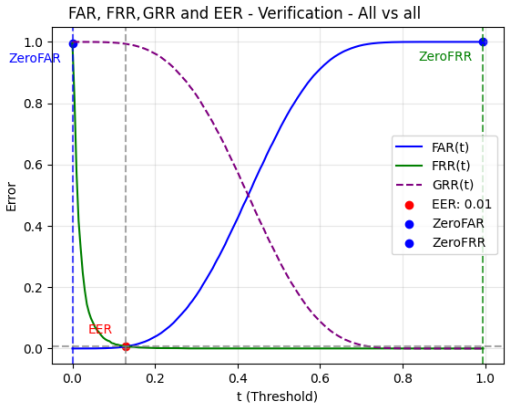

Figure 8: Verification All vs All

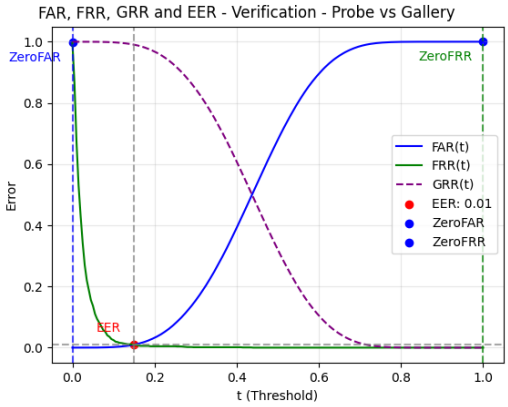

Figure 9: Verification Probe vs Gallery

These graphs provide a comprehensive evaluation of the normalized ResNet model's verification performance in both "All vs All" and "Probe vs Gallery" scenarios. It plots the False Rejection Rate (FRR) against varying thresholds, with key points labeled as ZeroFAR and ZeroFRR, marking thresholds where the False Acceptance Rate (FAR) and FRR are minimized to zero. The Equal Error Rate (EER) is recorded at 0.01, highlighting the model's high accuracy. This analysis aids in identifying the optimal threshold for balancing false acceptances and rejections, ensuring robust performance in diverse verification tasks.

4.4 Verification results

Method	EER	t@EER
Open-Iris	$\sim 0\%$	0.41
Full ResNet	1%	0.125
Norm. ResNet	1%	0.125

Table 1: Verification All vs all

Method	EER	t@EER
Open-Iris	$\sim 0\%$	0.41
Full ResNet	1%	0.145
Norm. ResNet	1%	0.145

Table 2: Verification Probe vs Gallery

4.5 Identification

For the identification task, a gallery of known individuals and a set of probe samples are used, and for each probe, the following steps are taken:

- Match it against all templates in the gallery.
- If the probe belongs to a known individual, ensure the correct identity is ranked first *(DIR@1)*.
- If the probe is from an unknown individual, check if the system correctly rejects it as *"unknown"*.

4.5.1 Open Iris Library

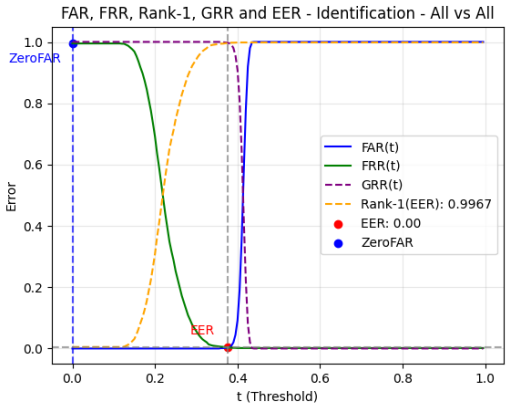

Figure 10: Identification All vs All

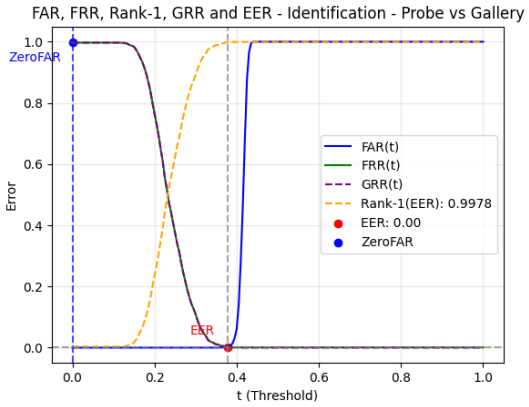

Figure 11: Identification Probe vs Gallery

These graphspresent the identification performance of the system in both "All vs All" and "Probe vs Gallery" scenarios. It plots key metrics, including the False Acceptance Rate (FAR), False Rejection Rate (FRR), Rank-1, Genuine Recognition Rate (GRR), and Equal Error Rate (EER), against varying thresholds. Critical points such as ZeroFAR and ZeroFRR are highlighted, marking thresholds where FAR and FRR are minimized or reach zero, providing insights into the system's effectiveness in different identification tasks.

4.5.2 Resnet-50 with full eye images

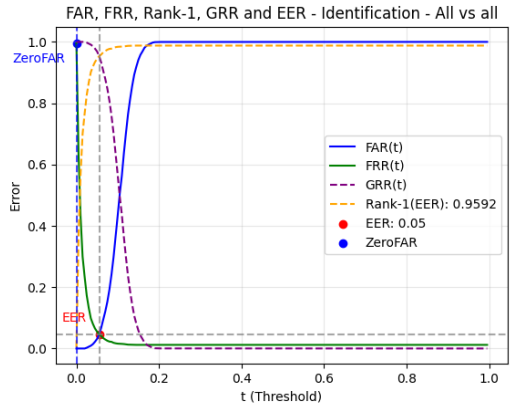

Figure 12: Identification All vs All

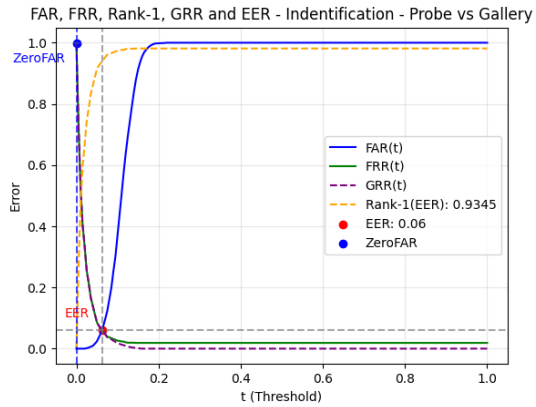

Figure 13: Identification Probe vs Gallery

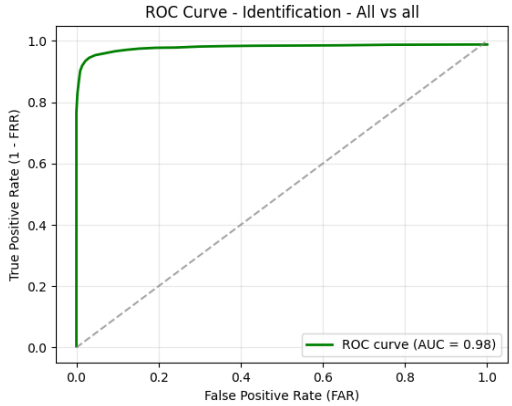

Figure 14: Identification ROC All vs All

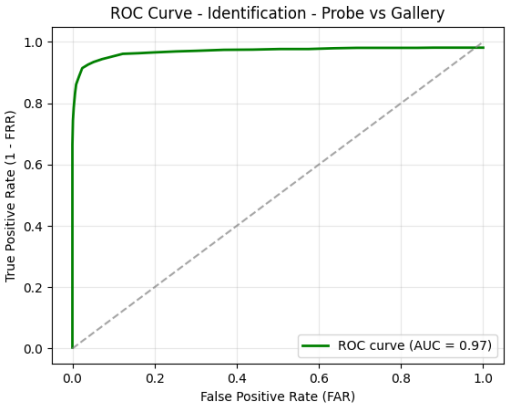

Figure 15: Identification ROC Probe vs Galley

This analysis evaluates the identification performance of the system across both All vs. All and Probe vs. Gallery configurations. The All vs. All setup achieves an Equal Error Rate (EER) of 0.05 and a Rank-1 accuracy of 95.92%, demonstrating excellent reliability and low error rates. In comparison, the Probe vs. Gallery configuration yields an EER of 0.06 and a Rank-1 accuracy of 93.45%, reflecting slightly lower but still strong performance. The accompanying Receiver Operating Characteristic(ROC) curves further assess the model's discriminative ability. The All vs. All ROC curve, with an Area Under the Curve (AUC) of 0.98, highlights superior performance, while the Probe vs. Gallery ROC curve achieves an AUC of 0.97, demonstrating robust identification capabilities in distinguishing between matching and non-matching pairs. The steepness of the curves and their proximity to the top-left corner emphasize the model's high effectiveness in both scenarios.

4.5.3 Resnet-50 with normalized eye images

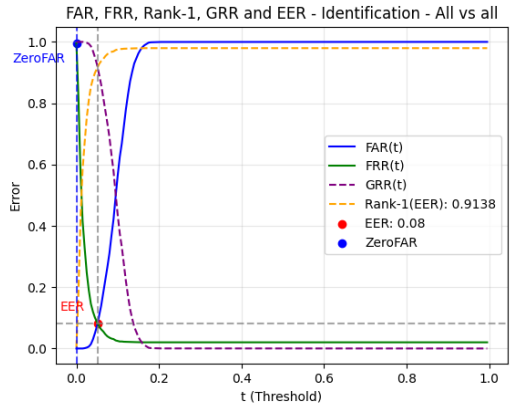

Figure 16: Identification All vs All

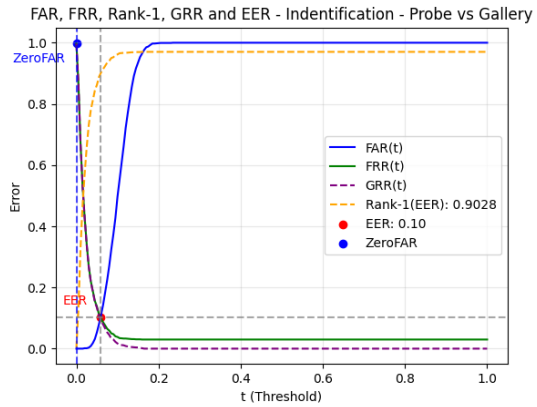

Figure 17: Identification Probe vs Gallery

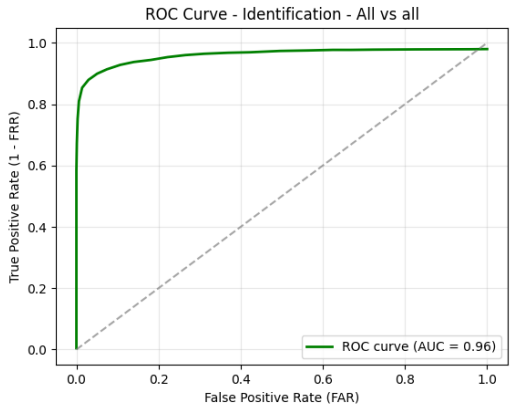

Figure 18: Identification ROC All vs All

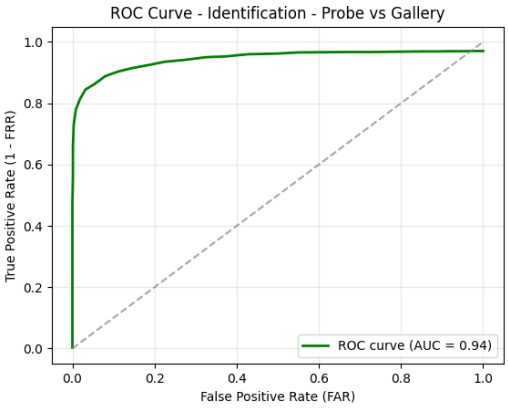

Figure 19: Identification ROC Probe vs Gallery

This analysis examines the identification performance of the normalized ResNet model in both ”All vs All” and ”Probe vs Gallery” scenarios. In the ”All vs All” configuration, the graph plots the False Acceptance Rate (FAR) and False Rejection Rate (FRR) against varying thresholds, with key points such as ZeroFAR highlighting thresholds where FAR is minimized to zero. The accompanying Receiver Operating Characteristic (ROC) curve demonstrates the model's ability to distinguish between positive and negative samples, achieving an Area Under the Curve (AUC) of 0.96, indicative of excellent performance.

In the ”Probe vs Gallery” scenario, the graph extends the analysis by including FAR, FRR, and additional metrics. It highlights critical points such as ZeroFAR and the Equal Error Rate (EER) of 0.10. The Rank-1 Error Rate (ERR) is noted as 0.9028, showcasing high accuracy. The corresponding ROC curve, with an AUC of 0.94, further underscores the model's strong discriminative ability in this context, effectively distinguishing between probe and gallery samples. These evaluations collectively emphasize the robust identification capabilities of the normalized ResNet model across diverse scenarios.

4.6 Identification results

Method	EER	t@EER	DIR@1	AUC
Open-Iris	$\sim 0\%$	0.375	99.7%	-
F. ResNet	5%	0.06	95.92%	98%
N. ResNet	8%	0.05	91.38%	96%

Table 3: Identification All vs All

Method	EER	t@EER	DIR@1	AUC
Open-Iris	$\sim 0\%$	0.375	99.8%	-
F. ResNet	6%	0.060	93.39%	97%
N. ResNet	10%	0.060	90.21%	94%

Table 4: Identification Probe vs Gallery

4.7 OpenIris Results Analysis

Using the OpenIris library for iris recognition, normalization, feature extraction, and matching on the CASIA-Iris-Thousand dataset, we achieved results that align closely with those reported by the authors of the library. Specifically, the Equal Error Rate (EER) threshold obtained during our evaluation was 0.375, which is remarkably close to the optimal threshold of 0.37 stated in the library's documentation. This consistency highlights the reliability and robustness of the OpenIris framework for iris recognition tasks.

The exceptional performance of OpenIris can be attributed to its specialized design and optimization specifically for iris recognition. Unlike general-purpose architectures such as ResNet, OpenIris employs domain-specific techniques that are finely tuned to handle the unique challenges of iris recognition. These include:

- **Advanced Preprocessing**: OpenIris applies sophisticated preprocessing techniques that effectively remove noise, reflections, and occlusions, while normalizing the iris region to a consistent scale. This ensures that the input images are of high quality, providing a strong foundation for accurate recognition.

- **Tailored Feature Extraction**: The library uses feature extraction methods specifically designed to capture the intricate radial and textural patterns of the iris. These features are critical for distinguishing between individuals, as they minimize intra-class variability and enhance inter-class separability.

- **Optimal Threshold Calibration**: OpenIris incorporates well-calibrated decision thresholds that maximize the separability between genuine and impostor matches. This is evident from its low EER threshold of 0.375, reflecting the library's ability to confidently distinguish between matching and non-matching iris pairs.

4.8 Comparative Analysis

The comparative analysis highlights the performance of Open-Iris, Full ResNet, and Normalized ResNet across verification and identification tasks. It exhibits a higher threshold at EER of 0.41, indicating stronger confidence in decision thresholds compared to Full ResNet and Normalized ResNet, which both achieve an EER of 1% and have lower threshold at EER values of 0.125 and 0.145, respectively. Full ResNet and Normalized ResNet demonstrate identical EER values but differ slightly in threshold at EER between scenarios, with both methods trailing behind Open-Iris in terms of overall performance.

In the identification results, Open-Iris also emerges as the superior method, achieving$\sim 0\%$ EER in both the "All vs All" and "Probe vs Gallery" scenarios. It achieves near-perfect identification rates with DIR@1 values of 99.7% and 99.8%, respectively, and maintains a strong threshold at EER (t@EER) of 0.375 in both cases. Full ResNet performs reasonably well, with EER values of 5% and 6%, DIR@1 values of 95.92% and 93.39%, and AUC values of 98% and 97% for "All vs All" and "Probe vs Gallery," respectively. However, its performance is noticeably weaker compared to Open-Iris. Normalized ResNet shows further performance degradation, with EER values of 8% and 10%, DIR@1 values of 91.38% and 90.21%, and AUC values of 96% and 94% across the two scenarios.

Overall, Open-Iris demonstrates the most robust and consistent performance across both verification and identification tasks, significantly outperforming the ResNet-based methods. While Full ResNet performs better than Normalized ResNet, both methods are less effective than Open-Iris, with higher error rates, lower recognition rates, and weaker thresholds.

5 Conclusion

This paper compares the traditional and deep learning-based approaches for iris recognition,

considering the segmentation, normalization, feature extraction, and matching stages. The classic pipeline implemented using the OpenIris library outperformed most of them, especially in terms of **Equal Error Rate (EER)** and **Rank-1 accuracy**. This underlines the reliability and strength of the classical methods, which are based on well-established techniques for feature extraction using Gabor filters and Hamming distances for matching. In this respect, these methods tend to do very well under controlled conditions with good lighting and cooperation of the users.

On the other hand, deep learning was done with ResNet-50 architecture and appeared more versatile against complex patterns in iris variations. Specifically, the ResNet-50 model returned competitive results when trained with normalized iris images: 1% EER in verification tasks and 91.38% Rank-1 accuracy in identification tasks. The effective ability to handle such challenging scenarios was provided to the deep learning model through automatic learning of hierarchical features from the data.

The identification task performed a bit worse by the deep learning model, especially within the "Probe vs Gallery" scenario where the EER is taken to 10%. This, therefore, might illustrate the fact that even though the deep learning methods are powerful, it could require further optimizations to match, in some contexts, the traditional methods' accuracy.

The results also pointed out how segmentation accuracy influences the overall performance of the iris recognition system. This was achieved using the dual-headed neural network architecture for segmentation in the OpenIris library, which was able to successfully segment the iris region from high-resolution infrared images with a success rate of 93%. This is very important because mistakes made in the segmentation step would have propagated through to the next steps and hence gave poor results in feature extraction and matching.

5.1 Future Improvements

While traditional and deep learning techniques showed excellent results, there are some points at which improvements might be made in future work to enhance the effectiveness of iris recognition systems.

1. **Improved Segmentation Methods**: The current segmentation method, while very effective, can be further improved by utilizing the latest neural network architectures, such as **transformer-based models** or **attention mechanisms**, that will improve the accuracy of iris localization in challenging conditions, such as occlusions or poor lighting.

2. **Hybrid Approaches**: Combining traditional and deep learning methods may result in a stronger model. The two approaches can combine their strengths, where classic feature extraction could be represented, for example, by Gabor filters, together with deep learning topologies, leading to an enhancement concerning the extracted discriminative characteristics of the representations.

3. **Larger and More Diverse Datasets**: Training deep learning models on larger and more diverse datasets, including images captured under varying lighting conditions, with different types of occlusions, and from a wider range of ethnicities, could improve the generalization capabilities of the models. This would make the system more reliable in real-world applications.

4. **Real-Time Processing**: The deep learning models should be optimized for real-time processing, which could be achieved by quantization or pruning of the models to prepare them for real-world deployment in security systems where low latency is important.

5. **Spoofing Detection**: Future work could be addressed to make the system more robust against spoofing, employing, for example, high-resolution printed images or artificial contact lenses. Anti-spoofing techniques integrated into the recognition pipeline would increase the security of the systems even more.

6. **User Experience**: The user experience for people of varying abilities or small children may lie in developing even friendlier interfaces or ways of capturing an iris without needing users to precisely comply.

From both, again traditional methods are strengths but also limited as those based on deep learning, the future is toward incorporating methods for segmentation and extraction along with continuous

anti-spoofing improvements. Therefore, addressing all these challenges leads to iris recognition systems that could get even more reliable, secure, and adaptable to almost any real-world setup or application.

5.2 Demo

The demo is a simple identification system built using PyQt5, which integrates the implementations of both the OpenIris library and the ResNet-50 models for iris recognition. The system is designed to register and identify users based on their iris patterns. A database was created to store user information, including a table for users and a related table that stores the iris embeddings for each model (OpenIris and ResNet-50). Each user can register multiple eye images, capturing different positions or angles, to ensure robustness during identification.

The application allows new users to register through a form where they provide their name and upload an eye image. During the identification process, the user submits another eye image, and the system generates the corresponding embeddings for that image. These embeddings are then compared with those stored in the database using Hamming distance for OpenIris embeddings and cosine similarity for ResNet-50 embeddings. The system produces a list of distances, ordered from the lowest (most likely match) to the highest (least likely match). To ensure balanced performance, the **Equal Error Rate (EER)** values were used as thresholds for determining whether a match is valid. This approach ensures a fair trade-off between false acceptances and false rejections, providing a reliable and user-friendly identification system.

References

[1] Daugman, J. G. (2004). How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 21–30. https://www.cl.cam.ac.uk/~jgd1000/irisrecog.pdf

[2] D. Gabor, Theory of communication. Part 1: The analysis of information https://digital-library.theiet.org/doi/10.1049/ji-3-2.1946.0074

[3] Hamming, R. (1950). Error Detection and Error Correction. Bell System Technical Journal, 29(2), 147–160. https://ieeexplore.ieee.org/document/6772729

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://ieeexplore.ieee.org/document/7780459

[5] Worldcoin AI (2023). IRIS: Iris Recognition Inference System of the Worldcoin project. https://github.com/worldcoin/open-iris

[6] Iris recognition inference system https://world.org/blog/engineering/iris-recognition-inference-system

[7] S. Ahmad and B. Fuller, "ThirdEye: Triplet Based Iris Recognition without Normalization," 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 2019, pp. 1-9, doi: 10.1109/BTAS46853.2019.9185998. https://ieeexplore.ieee.org/document/9185998

[8] Hoffer, E., & Ailon, N. (2015). Deep Metric Learning Using Triplet Network. In Proceedings of the Neural Information Processing Systems (NeurIPS). "Deep Metric Learning Using Triplet Network" https://arxiv.org/abs/1412.6622

[9] CASIA-Iris-Thousand. *Iris Biometric authentication model*, https://www.kaggle.com/datasets/sondosaabed/casia-iris-thousand