
A COMPARISON BETWEEN THE ENHANCING TECHNIQUES OF FINGERPRINT

Apostol Cristian

Sapienza, University of Rome
apostol.2002291@studenti.uniroma1.it

Costanzi Fantini Pietro

Sapienza, University of Rome
costanzifantini.1982805@studenti.uniroma1.it

ABSTRACT

Fingerprint enhancement is a crucial preprocessing step in fingerprint recognition. Despite being the most widely used and well-known biometric trait, extracting features from low-quality fingerprint images remains a significant challenge. The purpose of this paper is to develop a biometric system that integrates and evaluates various enhancement strategies. Specifically, we compare multiple enhancement techniques by implementing them within a fingerprint recognition pipeline and assessing their impact on feature extraction and overall recognition performance. Through this comparative analysis, we aim to identify the most effective approach for improving fingerprint recognition performance.

Keywords Fingerprint recognition · Enhancing · Minutiae · Singularities · Biometric System

1 Introduction

Fingerprint recognition is one of the most reliable and widely used biometric methods for personal identification and authentication. Over time, the technology has evolved significantly, with one of the most crucial preprocessing steps being fingerprint enhancement. This process plays a pivotal role in improving the quality of fingerprint images, particularly when dealing with noisy or low-resolution inputs. Various enhancement techniques have been developed over the years, each targeting different aspects of fingerprint image quality and feature extraction.

The foundational work in fingerprint enhancement was initiated by Hong et al. [1], whom developed techniques based on Gabor filters and frequency analysis to improve contrast and ridge structure. Hong's approach laid the groundwork for more advanced methods. Peter Kovesi [2] later expanded upon Hong's work, refining the estimation of the orientation field and improving robustness in handling noisy or degraded images. More recently, deep learning-based techniques, such as those proposed by Raffaele Cappelli [3, 4, 5, 6], have revolutionized fingerprint enhancement. These methods leverage convolutional neural networks (CNNs) to automatically learn complex features from fingerprint images, providing significantly improved performance over traditional methods. Moreover, Cappelli's research has introduced important considerations for evaluating performance at various stages of the enhancement process, particularly the creation of the orientation map of ridges and the estimation of frequencies.

This paper aims to compare and contrast different enhancement strategies, from the classical approaches to the most recent techniques not base on machine learning and deep learning. We deside to reinforce the old fashion techniques to have a very simple and straightforward biometric system. The paper is structured as follows: First, we provide an in-depth analysis of the common steps involved in fingerprint enhancement, including segmentation, orientation field estimation, frequency estimation, and filtering. Each of these steps will be discussed in detail, outlining the evolution of methods used over time. Following this, we present the construction of a fingerprint recognition system built upon these preprocessing steps. Finally, we evaluate the performance of the system, comparing the results of various enhancement techniques based on standard performance metrics.

2 Enhancing Pipeline

In this section, we describe the general approach behind the stages of fingerprint enhancement. These stages can be broadly divided into four steps: defining the Region of Interest (ROI), estimating the ridge orientations and their frequencies, and using the previous information to enhance the visibility of the ridges.

Each of these stages can be addressed using different algorithms and methodologies, each with its own advantages and limitations. Our goal is to analyze and compare multiple approaches for each stage, and evaluating their impact on the overall performance of the fingerprint recognition system (see Sec. 6).

2.1 Segmentation

A critical step in fingerprint recognition is segmentation, which involves isolating the fingerprint pattern (ROI) from the background (see Fig. 1). This step is essential as it removes noise and irrelevant information, enabling more accurate fingerprint recognition algorithms. Over the years, various segmentation techniques have been proposed. In this paper, we follow the KISS (Keep It Simple and Straightforward) principle, as adopted by Cappelli, to compare simple yet effective segmentation methods developed by Kovesi P. in his matlab code [2], and Cappelli R.[4].

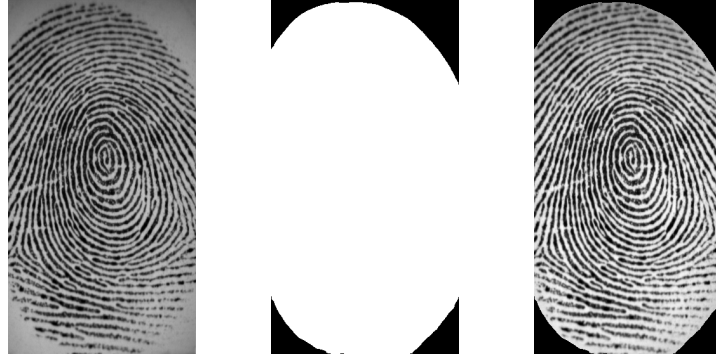


Figure 1: An example of fingerprint segmentation. (a) A fingerprint, (b) Segmentation mask, (c) Segmentation mask contour overlaid on the fingerprint.

2.1.1 Hong's segmentation

Hong distinguishes between recoverable and unrecoverable regions in fingerprint images by analyzing the shape of the wave formed by local ridges and valleys. In Hong's algorithm, this distinction is based on three key features that characterize the sinusoidal-shaped wave: amplitude (α), frequency (β), and variance (γ) (see Fig. 2).

[1] Let $X[1], X[2], \dots, X[l]$ be the x-signature of a block centered at (i, j) (see Sec. 2.4.1). The three features corresponding to pixel (block) (i, j) are computed as follows:

1. $\alpha = (\text{average height of the peaks} - \text{average depth of the valleys}).$
2. $\beta = \frac{1}{T(i, j)},$ where $T(i, j)$ is the average number of pixels between two consecutive peaks.
3. $\gamma = \frac{1}{l} \sum_{i=1}^l \left(X[i] - \frac{1}{l} \sum_{i=1}^l X[i] \right)^2$

Hong et al. also proposed a classification method to distinguish between recoverable and unrecoverable regions. Several fingerprint images were manually labeled, and the three extracted features were computed for each region, resulting in 2,000 three-dimensional patterns. A squared-error clustering algorithm was applied to identify six clusters, with four corresponding to recoverable regions and two to unrecoverable regions. These six cluster centers were then used as prototypes in a one-nearest neighbor (1NN) classifier to classify each $w \times w$ block of an input fingerprint image. If the percentage of recoverable regions in an image was below a threshold, the fingerprint was rejected.

This approach is particularly useful for rejecting poor-quality fingerprints during enrollment. However, the original work does not specify the dataset(s) used for gathering the training data, making fine-tuning of the classification process a challenging task. This highlights how preliminary steps in fingerprint enhancement, such as segmentation, require significant effort to achieve optimal performance.

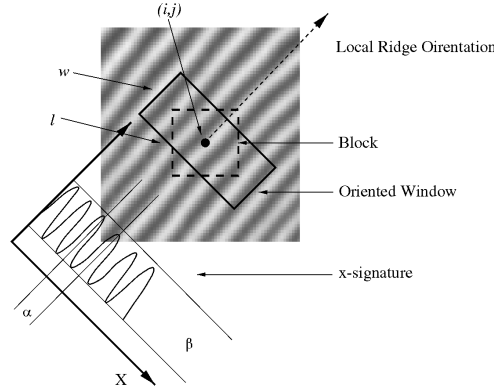


Figure 2: Sinusoidal wave from x-signature.[1]

The motivation behind presenting Hong’s method is twofold. First, it serves as a reference to highlight how the approaches adopted by Kovesi and Cappelli simplify segmentation by relying on thresholding techniques rather than complex feature-based classification. Second, it illustrates the shift in fingerprint processing pipelines over the years. While Hong’s method applies segmentation only after computing ridge orientations and frequencies, modern approaches prioritize segmentation in the early stages of enhancement. Notably, the x-signature used in Hong’s method for segmentation will reappear later in the frequency estimation stage (see Sec. 2.4), reinforcing the interdependence of these enhancement steps.

2.1.2 Kovesi’s segmentation

While Hong’s segmentation method involves multiple parameters and a detailed analysis of the ridge structure. Kovesi’s method adheres to the KISS principle by employing a single threshold-based criterion, avoiding the need for complex wave analysis. This makes the method more straightforward to understand and implement, while maintaining effective segmentation in almost the case.

In a fingerprint image, background regions generally exhibit very low gray-scale variance, whereas foreground regions have significantly higher variance. This method begins by dividing the image into blocks and calculating the gray-scale variance for each of them. If the variance is below a predefined global threshold, the block is classified as part of the background; otherwise, it is assigned to the foreground [7].

The gray-level variance for a block of size $W \times W$ is defined as:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (I(i, j) - M(k))^2 \quad (1)$$

where $V(k)$ is the variance for block k , $I(i, j)$ is the grey-level value at pixel (i, j) , and $M(k)$ is the mean grey-level value for the block k .

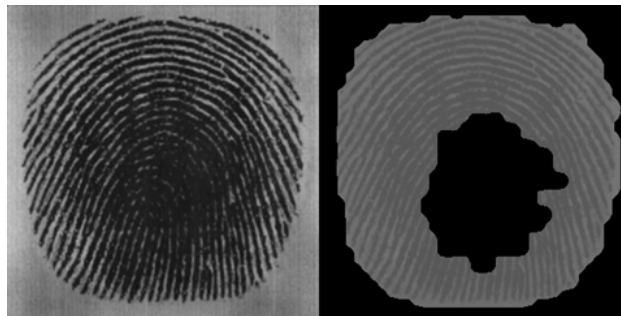


Figure 3: Failed segmentation from Kovesi’s pipeline.

However, this method is not always reliable, as it may produce segmentation results with holes or artifacts within the fingerprint region. Unlike Cappelli's method, which includes a dedicated post-processing step to refine the segmentation mask, Kovesi's approach does not guarantee a seamless foreground mask. An example of a failed segmentation result is shown in Fig. 3.

This issue is particularly noticeable in images captured with **capacitive sensors**, where certain regions of the fingerprint (especially those with uniform pressure) tend to appear darker and less defined. Unlike **optical sensors**, which capture ridges based on light reflection and absorption, capacitive sensors rely on electrical conductivity differences between ridges and valleys. When finger pressure is too high or the sensor surface conditions vary, ridges may appear less prominent, leading to a lower contrast and variance in the captured image. As a result, Kovesi's variance-based segmentation may incorrectly discard these regions, impacting the overall fingerprint recognition process.

2.1.3 Gradient Magnitude Filtering Segmentation (GMFS)

The GMFS method, proposed by Cappelli R. [4], was designed with three main objectives:

- It should rely on a small number of simple features (ideally only one).
- It should consist of a short sequence of well-known and efficient image processing steps.
- It should require only a few parameters to be configured.

The core of GMFS is based on the estimation of the **gradient magnitude** at each pixel. Given an input fingerprint image F , the horizontal and vertical derivative approximations of F , denoted as $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$, are computed using convolution with Sobel filters S_x and S_y :

$$\frac{\partial F}{\partial x} = F * S_x, \quad \frac{\partial F}{\partial y} = F * S_y \quad (2)$$

The gradient magnitude M is then computed as:

$$M = \sqrt{\left(\frac{\partial F}{\partial x}\right)^2 + \left(\frac{\partial F}{\partial y}\right)^2} \quad (3)$$

Since the gradient magnitude is typically high at ridge-valley transitions (see Fig. 4.b), a threshold t is computed to separate the fingerprint foreground from the background:

$$t = \text{percentile}(M, 95) \cdot \tau \quad (4)$$

where $\text{percentile}(M, 95)$ is the 95th percentile of matrix M and τ is a parameter of the method. Using the 95th percentile instead of the maximum value mitigates the impact of noise-induced outliers.

To further reduce noise and improve segmentation, the gradient magnitude is smoothed using a Gaussian filter G_σ :

$$\bar{M} = M * G_\sigma \quad (5)$$

The Gaussian filter G_σ is defined as:

$$G_{2D}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6)$$

It is discretized over a $g_s \times g_s$ grid, where $g_s = \lceil 3\sigma \rceil \cdot 2 + 1$, ensuring that most of the Gaussian values are contained according to the three-sigma rule¹(see Fig. 4.c).

The initial segmentation mask S_t is obtained by applying a thresholding operation on the smoothed gradient magnitude:

¹The "three-sigma rule," also known as the 68-95-99.7 rule, is a statistical guideline that applies to a normal distribution (also known as a Gaussian distribution). It describes the percentage of data points that fall within a certain number of standard deviations from the mean of the data.

$$S_t(i, j) = \begin{cases} \text{foreground,} & \text{if } \bar{M}_{i,j} > t, \\ \text{background,} & \text{otherwise.} \end{cases} \quad (7)$$

Note that t is computed from the non-averaged gradient magnitude M , while thresholding is applied to the averaged gradient magnitude $\bar{M}$ (see Fig. 4.d).

To refine the segmentation mask and enhance the quality of the segmentation, the GMFS method goes beyond the basic thresholding by applying a series of postprocessing steps. These steps are crucial in overcoming the limitations of simpler thresholding-based methods, such as those used in Kovesi's approach, and help to better handle noise and imperfections in the segmented image. The postprocessing steps include:

- **Morphological closing:** A dilation followed by erosion using a 3×3 disk-shaped structuring element, repeated n_c times. (see Fig 4e)
- **Largest connected component selection:** If the foreground contains multiple components, only the largest is retained. (see Fig 4f)
- **Hole filling:** Any holes not adjacent to the image border are filled. (see Fig 4g)
- **Morphological opening:** An erosion followed by dilation, repeated n_o times. (see Fig 4h)
- **Final cleanup:** If the previous step creates multiple components, the largest is selected again.

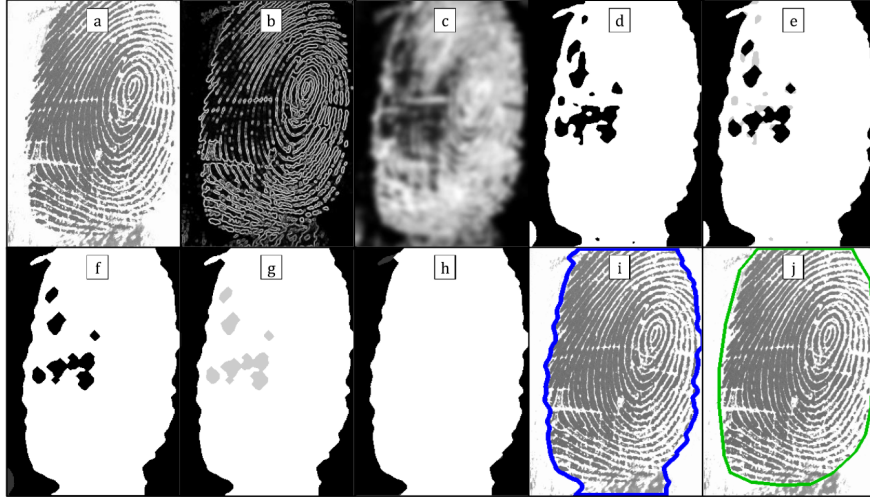


Figure 4: Illustrate the steps of GMFS method: (a) original image, (b) computed gradient magnitude of the image, (c) smoothed gradient magnitude with Gaussian filter, (d) thresholding operation on the average gradient magnitude, (e) noise-induced holes are partially filled, (f) bottom left artifact is removed, (g) remaining holes are filled and (h) small protrusion is eliminated. (i) Final segmentation mask, and (j) Ground truth.[4]

The GMFS method meets its design goals:

- It is based on a single feature: the **gradient magnitude**.
- It consists of efficient, well-known image processing operations: convolutions, thresholding, morphology operations, and connected component labeling.
- It requires only four parameters: τ , σ , n_c , and n_o , which have been set according to the values provided by Cappelli in his paper, based on the datasets he used for testing.

As it was mention before, the GMFS method was thoroughly tested using publicly available fingerprint databases from the first three Fingerprint Verification Competitions (FVC2000, FVC2002, and FVC2004), which are widely regarded as standard benchmarks for fingerprint comparison algorithms. These databases, along with the manually marked segmentation ground truth made available by Thai, Huckemann, and Gottschlich, have become essential for evaluating fingerprint segmentation methods. (Dataset on Kaggle)

Cappelli’s work utilized these databases to fine-tune the parameters of the GMFS method, ensuring that it performs optimally across various fingerprint qualities. In our work, we will particularly focus on testing the FVC 2002 databases, which will also be used in the evaluation phase of the recognition system discussed in Section 6.

2.2 Fingerprint Orientation Field

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint (see Fig. 5). Orientation estimation is a fundamental step in the enhancement process, as the subsequent filtering stage relies on the local orientation to effectively enhance the fingerprint image. Moreover, it plays a crucial role in feature extraction, particularly in detecting minutiae and singularities, which will be covered in Section 3 and 5.

Both Kovesi and Cappelli approaches follow a gradient-based method to estimate the orientation of the ridges, avoiding the explicit division into blocks typical of Hong’s method [1]. However, while Kovesi formalizes the problem in terms of gradient moments, Cappelli adopts a more direct formulation based on the weighted average of the squared gradients. In this section, we analyze both techniques in detail and highlight their differences.

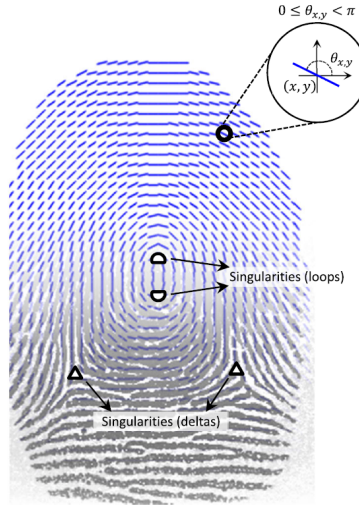


Figure 5: Orientation map.[5]

2.2.1 Kovesi’s Orientation Estimation

Given a normalized fingerprint image², the algorithm proceeds as follows:

1. **Gradient Computation:** The gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ are computed at each pixel (i, j) with the derivative of a *Gaussian* filter. The derivatives of the Gaussian kernel in the x and y directions are used to compute the image gradients:

$$\partial_x = I * G_x, \quad \partial_y = I * G_y \quad (8)$$

where G_x and G_y are the derivatives of the Gaussian kernel.

2. **Gradient Moments Computation:** To estimate the orientation field, the second-order gradient moments are computed as follows:

$$M_{x^2} = \partial_x^2, \quad M_{y^2} = \partial_y^2, \quad M_{xy} = \partial_x \cdot \partial_y \quad (9)$$

These moments are crucial because they provide a measure of the dominant gradient direction in each region of the image. To enhance robustness and reduce the influence of local noise, these moments are smoothed using a Gaussian filter with standard deviation σ_b :

$$M'_{x^2} = M_{x^2} * G_b, \quad M'_{y^2} = M_{y^2} * G_b, \quad M'_{xy} = 2(M_{xy} * G_b) \quad (10)$$

²Image normalization is a preprocessing step that adjusts the gray-level values of an image to reduce variations along the ridges and valleys, making it easier to extract ridge orientation.

This step ensures that small variations in the local gradient field do not result in abrupt changes in orientation estimation.

3. **Orientation Field Computation:** The local orientation $\theta(i, j)$ is determined analytically using the smoothed gradient moments. The angles are double because, in this way, opposite gradients reinforce each other, while perpendicular gradients cancel out.

$$\begin{aligned}\sin(2\theta) &= \frac{M'_{xy}}{\sqrt{M'^2_{xy} + (M'^2_{x^2} - M'^2_{y^2})^2}} \\ \cos(2\theta) &= \frac{(M'_{x^2} - M'_{y^2})}{\sqrt{M'^2_{xy} + (M'^2_{x^2} - M'^2_{y^2})^2}}\end{aligned}\quad (11)$$

The final orientation is then given by:

$$\theta(i, j) = \frac{\pi}{2} + \frac{1}{2} \arctan2(\cos(2\theta), \sin(2\theta)) \quad (12)$$

4. **Smoothing of the Orientation Field:** Since the estimated orientation field can be noisy due to fingerprint artifacts, minutiae, or ridge corruption, a final smoothing step is applied. This ensures a more coherent orientation field that better represents the overall ridge flow. The approach consists of convolving $\cos(2\theta)$ and $\sin(2\theta)$ with a Gaussian filter:

$$\begin{aligned}S_x(i, j) &= \cos(2\theta) * G_s \\ S_y(i, j) &= \sin(2\theta) * G_s\end{aligned}\quad (13)$$

The final smoothed orientation is then recomputed as:

$$\theta(i, j) = \frac{\pi}{2} + \frac{1}{2} \arctan2(S_y(i, j), S_x(i, j)) \quad (14)$$

While Kovesi's method provides a robust estimation of the orientation field through gradient moments and a fixed Gaussian smoothing, Cappelli introduces additional preprocessing steps and an adaptive smoothing strategy to further improve robustness, especially in noisy regions.

2.3 GBFOE (Gradient-Based Fingerprint Orientation Estimation)

In this section, we present a detailed explanation of the GBFOE method for fingerprint orientation estimation, emphasizing its mathematical formulation and key differences from Kovesi's approach. While both methods rely on gradient-based orientation estimation, they differ in how they process and aggregate gradient information. GBFOE incorporates a preprocessing step to enhance contrast and reduce noise before computing the orientation field, whereas Kovesi's method directly computes the orientation using gradient moments. Additionally, GBFOE dynamically adjusts the Gaussian smoothing parameter based on local orientation coherence, allowing for adaptive filtering that preserves details in high-quality regions while improving robustness in noisy areas. For a more comprehensive discussion, refer to [5], [8] and [9].

The first step in GBFOE is preprocessing the fingerprint image F using the segmentation mask S . This preprocessing includes:

1. **Contrast Stretching:** The extreme gray level values are clipped based on a specified percentage, denoted by sp , to enhance image contrast. Let H be the gray level histogram of the image F . The lower and upper bounds for clipping are determined by the gray levels L and U such that:

$$\sum_{i=0}^L H(i) \approx sp \cdot N, \quad \sum_{i=U}^{255} H(i) \approx sp \cdot N \quad (15)$$

where N is the total number of pixels in the image. The pixel intensities are then stretched to the full dynamic range $[0, 255]$.

2. **Gaussian Smoothing:** A 2D Gaussian filter with standard deviation σ_p is applied to the contrast-stretched image to reduce noise:

$$F' = F * G_{\sigma_p} \quad (16)$$

where G_{σ_p} is a 2D Gaussian kernel with standard deviation σ_p .

3. **Median Filtering:** A median filter with a filter size of m_p is applied to further reduce noise:

$$F'' = \text{median}_{m_p}(F') \quad (17)$$

where median_{m_p} denotes the median filtering operation with a window size of m_p . The resulting preprocessed image is denoted as F'' .

Gradient Averaging and Orientation Estimation

The local gradients of F'' are computed using Sobel operators, similar to Kovesi's implementation. Let $\partial_x(i, j)$ and $\partial_y(i, j)$ be the horizontal and vertical gradients at pixel (i, j) , respectively. The squared gradients are then computed:

$$G_x^2 = \partial_x^2, \quad G_y^2 = \partial_y^2, \quad G_{xy} = 2 \cdot \partial_x \cdot \partial_y \quad (18)$$

Kovesi's formulation in terms of gradient moments is conceptually equivalent to the GBFOE transformation, where the squared gradient terms G_x^2, G_y^2, G_{xy} play the same role. Both methods use a squared formulation to ensure that opposite gradients reinforce each other rather than cancel out, thereby improving the robustness of the orientation estimation. Like Kovesi, GBFOE smooths these terms using a Gaussian filter before computing the final orientation field.

These squared gradients are smoothed using a 2D Gaussian filter with standard deviation $\hat{\sigma}$:

$$G_x'^2 = G_x^2 * G_{\hat{\sigma}}, \quad G_y'^2 = G_y^2 * G_{\hat{\sigma}}, \quad G_{xy}' = G_{xy} * G_{\hat{\sigma}} \quad (19)$$

The local orientation $\theta(i, j)$ at each pixel (i, j) is then computed as:

$$\theta(i, j) = \frac{1}{2} \arctan 2(G_{xy}', G_x'^2 - G_y'^2) + \frac{\pi}{2} \quad (20)$$

Local Orientation Coherence and Final Estimation

To improve the robustness of the orientation estimation, a measure called local orientation coherence (Coh) is computed. The coherence at each pixel is calculated as:

$$Coh(i, j) = \frac{\sqrt{(G_x' - G_y')^2 + 4G_{xy}'^2}}{G_x' + G_y'} \quad (21)$$

The standard deviation σ of the Gaussian filter used in the final orientation estimation is dynamically adjusted based on the local orientation coherence, inside the segmentation mask S . This ensures that in regions where the fingerprint ridges are well-defined, less smoothing is applied to preserve fine details, while in noisy or low-quality regions, stronger smoothing is applied to improve stability.

$$\sigma = k_{\sigma} \left(1 - \frac{\sum_{i,j} (S_{i,j} \cdot Coh_{i,j})}{\sum_{i,j} S_{i,j}} \right) \quad (22)$$

Finally, the second smoothed orientation field is obtained by convolving the smoothed squared gradients with a Gaussian filter G_{σ} with the adjusted standard deviation:

$$G_x''^2 = G_x'^2 * G_{\sigma}, \quad G_y''^2 = G_y'^2 * G_{\sigma}, \quad G_{xy}'' = G_{xy}' * G_{\sigma} \quad (23)$$

The final orientation field is then:

$$\theta(i, j) = \frac{1}{2} \arctan 2(G_{xy}'', G_x''^2 - G_y''^2) + \frac{\pi}{2} \quad (24)$$

Parameters

The method is controlled by five main parameters:

- sp : the contrast stretching clipping percentage,
- σ_p : the standard deviation of the Gaussian smoothing filter,
- m_p : the size of the median filter,
- $\hat{\sigma}$: the standard deviation of the Gaussian filter for the first orientation estimation,
- k_σ : a multiplicative factor to adjust σ in the final smoothing.

These parameters influence the accuracy of the estimated orientation field and, consequently, the overall performance of the fingerprint recognition system built upon this method. To optimize the parameter selection, Cappelli R. conducted a fine-tuning process using datasets distinct from the FVC benchmarks, ensuring that the final evaluation remains unbiased.

Cappelli’s fine-tuning approach involved testing the method on a variety of fingerprint datasets captured with different sensors and under varying quality conditions. The optimization focused on minimizing orientation estimation errors, particularly in low-quality fingerprint regions where ridges are fragmented or corrupted by noise. The evaluation metric used in this process was the **Root Mean Square Deviation (RMSD)** between the estimated and ground truth orientation fields. RMSD is computed as follows:

$$RMSD(\theta, \hat{\theta}, S) = \sqrt{\frac{\sum_{i,j} S_{i,j} \cdot d\phi(\theta_{i,j}, \hat{\theta}_{i,j})^2}{\sum_{i,j} S_{i,j}}} \quad (25)$$

where θ represents the estimated orientation field, $\hat{\theta}$ is the ground truth, S is the ground truth segmentation mask, and $d\phi(\theta, \hat{\theta})$ is the angular difference:

$$d\phi(\theta, \hat{\theta}) = (\theta - \hat{\theta} + \frac{\pi}{2}) \mod \pi - \frac{\pi}{2} \quad (26)$$

By leveraging this metric, the fine-tuning process aimed to ensure robust orientation estimation across various fingerprint datasets. However, it is important to emphasize that the FVC datasets were not used in this tuning phase. These datasets will instead be employed exclusively in the final evaluation (Section 6) of our fingerprint recognition system, allowing for an objective assessment of its generalization capabilities.

2.4 Ridge Frequency Estimation

The ridge frequency represents the local spacing between fingerprint ridges and is typically measured in terms of the number of ridges per unit distance. Various methods have been proposed, each with different approaches to handling noise, singular points, and irregular ridge patterns.

This section presents two methods for ridge frequency estimation: Kovesi’s method, which is based on compute a mean global frequency and XSFFE, a more recent approach that leverages x-signatures to achieve efficient and robust frequency estimation proposed by Cappelli [6].

2.4.1 Kovesi’s frequency estimation

In regions where neither minutiae nor singular points are present, the grayscale levels along the ridges and valleys can be approximated as a sinusoidal wave perpendicular to the local ridge orientation (see Fig 2). Thus, the local ridge frequency is an inherent characteristic of fingerprint images. Given a normalized image I and an orientation image Θ , the ridge frequency estimation follows these steps:

1. Divide the image I into non-overlapping blocks of size $w \times w$ (typically 16×16 pixels).
2. For each block centered at pixel (i, j) , define an oriented window with the same dimensions aligned with the ridge orientation, taken from the orientation map Θ .

3. Compute the x-signature $X[k]$ of ridges and valleys within the oriented window:

$$X[k] = \sum_{d=0}^{w-1} I(u, v) \quad (27)$$

where the coordinates (u, v) of the oriented window are given by:

$$\begin{aligned} u &= i + (d - \frac{w}{2}) \cos(\Theta(i, j)) - (k - \frac{1}{2}) \sin(\Theta(i, j)) \\ v &= j + (d - \frac{w}{2}) \sin(\Theta(i, j)) + (\frac{1}{2} + k) \cos(\Theta(i, j)) \end{aligned} \quad (28)$$

If no singular points or minutiae are present, the x-signature resembles a discrete sinusoidal wave, whose frequency corresponds to that of the ridges and valleys in the block.

4. The ridge frequency $W(i, j)$ is determined by $T(i, j)$, the average number of pixel between two consecutive peaks in the x-signature:

$$W(i, j) = \frac{1}{T(i, j)} \quad (29)$$

For a fingerprint image scanned at a fixed resolution, the value of the frequency of the ridges and valleys in a local neighborhood lies in a certain range. For a 500dpi image (all the databases used for the evaluation in Section 6 use this resolution), this range is $[1/3, 1/25]$. Therefore, if the estimated value of the frequency is out of this range, then the frequency is assigned a value of 0 to indicate that a valid frequency cannot be obtained. In this manner, the mean global frequency considered only the well-defined regions for the computation.

Despite its extreme simplicity, this method introduces significant compromises, penalizing particularly sensitive areas such as those adjacent to minutiae and singularities, which are characterized by structural deformations. Furthermore, the assumption of a constant distance between ridges is not always valid; different regions of the finger can experience varying pressures during acquisition, leading to non-linear deformations.

2.4.2 X-Signature-based Fingerprint Frequency Estimation (XSFFE)

[6] XSFFE (X-Signature-based Fingerprint Frequency Estimation) is a spatial-domain method that relies on traditional image processing techniques and the computation of the x-signature. It follows the KISS principle, consisting of a straightforward sequence of well-established image processing operations, easily implemented using standard computer vision libraries such as OpenCV.

The process begins with a preprocessing step, where median filtering and Gaussian smoothing are applied to reduce noise and generate smoother x-signatures. The x-signature is then computed within a 23×43 window, considering only foreground pixels that are at least 11 pixels away from the background. This constraint ensures that low-quality regions near the background, which might lead to inaccurate estimations, are excluded. The choice of these parameters is crucial: a window that is too small becomes susceptible to local noise, such as small cuts or abrasions on the fingertip, whereas a window that is too large reduces accuracy in high-curvature areas. Similarly, setting a proper distance from the background helps to exclude noisy regions, but an excessive value may negatively impact accuracy.

For each analyzed pixel, both minima and maxima of its x-signature are examined. The distances between consecutive maxima represent the spacing between valleys, while the distances between consecutive minima indicate the spacing between ridges³. By incorporating both types of distance for the discrimination, the method achieves a more robust estimation. If the number of local maxima and minima is at least a specified threshold, then the ridge frequency analysis is performed. Furthermore, outlier distances between consecutive peaks / valleys (filtered with a minimum and maximum period) are identified by first computing the median distance and then discarding any value differing by more than two pixels from the median. If at least four valid distances remain, the ridge frequency is estimated as the inverse of their average.

In cases where a frequency value cannot be reliably computed, either due to proximity to the background or an insufficient number of valid distances, missing values are determined using an inpainting technique. Finally, a Gaussian filter is applied to smooth the frequency map, ensuring a consistent and refined output.

As Cappelli also pointed out, there have been very few advances in this field in recent years, likely due to the difficulty in constructing datasets that contain true ridge frequencies. This process is typically performed manually by experts. In

³Cappelli's analysis is performed on a grayscale image where the highest pixel values correspond to white regions (valleys) and the lowest to dark regions (ridges).

[6], a method is also presented for constructing datasets that include ground truth segmentation masks, true orientation fields, and true ridge frequencies, stored in four grayscale PNG images. This dataset would greatly support future research, enabling much more accurate evaluations of each proposed method.

A metric introduced specifically to measure the accuracy of frequency estimation is the **Mean Absolute Percentage Error (MAPE)**. This choice is motivated by the practical significance of frequency estimation errors in subsequent fingerprint processing steps:

$$MAPE(\mathcal{F}, \hat{\mathcal{F}}, S) = \frac{\sum_{i,j} \left| \frac{1}{\mathcal{F}_{i,j}} - \frac{1}{\hat{\mathcal{F}}_{i,j}} \right| \cdot \hat{\mathcal{F}}_{i,j} \cdot S_{i,j}}{\sum_{i,j} S_{i,j}} \quad (30)$$

where $\mathcal{F}$ represents the estimated frequency map, $\hat{\mathcal{F}}$ denotes the ground truth frequency map, and S is the ground truth segmentation mask ($S_{i,j}$ is one for foreground pixels, zero otherwise).

2.5 Filtering

After segmentation, orientation estimation, and frequency estimation, filtering is the final step that refines the fingerprint image before further processing. In this section, we discuss two widely used filtering techniques: traditional Gabor filtering used in Hong et al. and GBFEN (Gabor-Based Fingerprint Enhancement) by Raffaele Cappelli. While both methods leverage Gabor filters to enhance ridge-valley patterns, GBFEN introduces refinements that improve adaptability and overall effectiveness, particularly in challenging fingerprint datasets.

2.5.1 Gabor Filtering

The Gabor filter is a powerful tool for fingerprint image enhancement, based on a sinusoidal plane wave modulated by a Gaussian envelope. This filter is particularly effective because of its ability to selectively respond to specific orientations and frequencies, which is key for preserving ridge structures while reducing noise. The Gabor filter used in this method is even-symmetric, and its real part is defined as a cosine wave modulated by a Gaussian function.

The even-symmetric Gabor filter in the spatial domain is defined as:

$$G(x; y; \mu; f) = \exp \left(-\frac{1}{2} \left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right] \right) \cos(2\pi f x_\mu) \quad (31)$$

where $x_\mu = x \cos \mu + y \sin \mu$ and $y_\mu = -x \sin \mu + y \cos \mu$ are the rotated coordinates in the filter's local coordinate frame, and μ is the orientation of the Gabor filter, and f is the frequency of the cosine wave. σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively.

The Gabor filter is applied by convolving the fingerprint image with the filter. For each pixel (i, j) in the image, the corresponding orientation value $O(i, j)$ and ridge frequency value $F(i, j)$ are used to calculate the filter response. The enhancement of the image $E(i, j)$ at each pixel is given by:

$$E(i, j) = \sum_{u=-w_x/2}^{w_x/2} \sum_{v=-w_y/2}^{w_y/2} G(u, v; O(i, j), F(i, j)) N(i - u, j - v) \quad (32)$$

where $O(i, j)$ and $F(i, j)$ are the orientation and ridge frequency images, respectively, N is the normalized fingerprint image, and w_x and w_y are the width and height of the Gabor filter mask.

The bandwidth of the filter, which specifies the range of frequencies it responds to, is determined by the standard deviation parameters σ_x and σ_y . In the original algorithm by Hong et al., these values were empirically set to fixed values of $\sigma_x = 4.0$ and $\sigma_y = 4.0$. However, this approach assumes a constant bandwidth, which can lead to suboptimal performance when ridge frequency varies significantly across the image.

To address this, Kovesi adapts the values of σ_x and σ_y according to the local ridge frequency $F(i, j)$, as follows:

$$\sigma_x = k_x / F(i, j) \quad \sigma_y = k_y / F(i, j) \quad (33)$$

where k_x and k_y are constant scaling factors, making the filter more adaptable to varying ridge frequencies across the image.

In the original method of Hong et al., the width and height of the filter mask were fixed at 11 pixels. However, this static size may not accommodate Gabor waveforms of different bandwidths. To allow the filter size to vary with the bandwidth, Kovesi define the filter mask size as a function of σ_x and σ_y :

$$w_x = 6\sigma_x \quad w_y = 6\sigma_y \quad (34)$$

This ensures that the filter size is large enough to capture the majority of the Gabor waveform information, which is typically contained within the region $[-3\sigma, 3\sigma]$ (three sigma rule).

By adjusting the filter parameters based on the local ridge frequency, this method offers a more adaptable and effective approach for fingerprint image enhancement, particularly in images where ridge frequencies are not uniform.

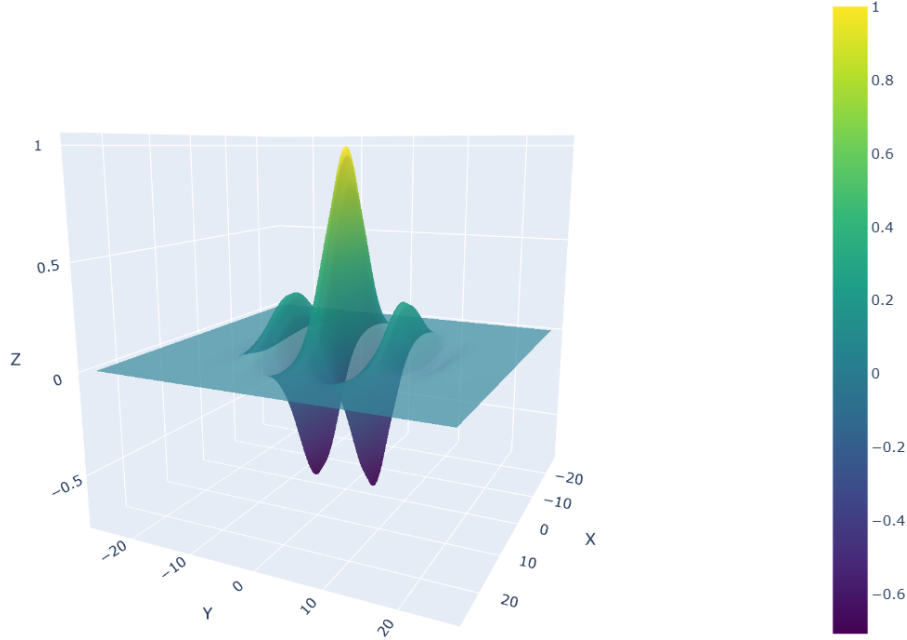


Figure 6: Plot of Gabor filter with $k_x = 0.85$, $k_y = 0.85$ and a frequency of 0.1

2.5.2 Gabor-Based Fingerprint Enhancement (GBFEN)

GBFEN is a simple and effective fingerprint enhancement method developed by Raffaele Cappelli. It is based on traditional contextual convolution using Gabor filters. The method has been shown to outperform several state-of-the-art techniques, particularly when combined with SNFOE (Single-Scale Normalized Fingerprint Orientation Estimation) and SNFFE (Single-Scale Normalized Fingerprint Frequency Estimation). In fact, GBFEN surpasses the performance of existing methods on the challenging NIST SD27 latent fingerprint database [10].

Although the paper that describes GBFEN in detail is not yet available, its effectiveness is highlighted in several comparative studies. The method is particularly valued for its simplicity and the ability to adapt the Gabor filter parameters for improved ridge and valley detection, making it an excellent choice for practical fingerprint enhancement tasks.

Given an input image $I(x, y)$, the enhancement function can be expressed as:

$$I'(x, y) = I(x, y) * G_{\theta, \lambda}(x, y) \quad (35)$$

where $G_{\theta, \lambda}(x, y)$ represents the Gabor filter with orientation θ and ridge period λ , and $*$ denotes the convolution operation. The values of θ and λ are determined from the orientation and frequency fields of the fingerprint image.

A crucial aspect of GBFEN is the discretization of the ridge period and orientation into a set of representative values. This allows for efficient selection of the most appropriate filter without performing continuous parameter adaptation. The discretization functions can be defined as:

$$\lambda_d = \text{round} \left(\frac{\lambda - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} (N_\lambda - 1) \right) \quad (36)$$

$$\theta_d = \text{round} \left(\frac{\theta}{\pi} N_\theta \right) \bmod N_\theta \quad (37)$$

where N_λ and N_θ are the total number of discrete values for ridge period and orientation, respectively. The enhancement is then performed by applying the precomputed filter indexed by (θ_d, λ_d) to the corresponding region of the image.

This method ensures that ridge structures are effectively enhanced while noise is suppressed. However, the performance of GBFEN strongly depends on the accuracy of the orientation and frequency estimation. Errors in these steps can lead to distortions in the enhanced image, necessitating additional post-processing techniques for refinement.

For further information regarding SNFOE and SNFFE, please refer to the references [4] and [5].

Filter Bank Construction: Differences between Kovesi and GBFEN

While both Kovesi's and GBFEN's methods rely on Gabor filtering, they differ in how the filters are generated and applied. The main differences lie in the estimation of ridge frequency, the selection of the appropriate filter, and the overall construction of the filter bank.

Definition of σ_x and σ_y

A key difference between the two approaches is how the parameters σ_x and σ_y , which control the Gaussian envelope of the filter, are defined:

- **Kovesi's Method:** The filter bank is constructed based on a **global** ridge frequency estimate, denoted as f_{global} . The values of σ_x and σ_y are computed as:

$$\sigma_x = \frac{k_x}{f_{\text{global}}}, \quad \sigma_y = \frac{k_y}{f_{\text{global}}} \quad (38)$$

where k_x and k_y are fixed scaling parameters. This implies that the filter bank is designed for a single frequency, without adapting to local variations in ridge frequency.

- **GBFEN Method:** In contrast, GBFEN uses a **local** ridge frequency estimate $f(i, j)$, computed for each region of the fingerprint image. The σ parameters are then adapted accordingly:

$$\sigma = \frac{5p}{12} \quad (39)$$

where p represents the local ridge period. This allows the filter to better adapt to spatial variations in ridge frequency.

Filter Selection

Another fundamental difference lies in how the appropriate filter is selected for each region of the image:

- **Kovesi's Method:** The filter bank consists of a set of Gabor filters that are dynamically rotated to match the estimated ridge orientation at each pixel. The orientations are continuous, meaning that each filter is generated on-the-fly at the exact estimated orientation $O(i, j)$.
- **GBFEN Method:** Instead of dynamically rotating filters, GBFEN precomputes a set of filters and selects the closest one based on a discretized orientation index θ_d and a discretized frequency index λ_d . This approach reduces computational complexity at the cost of a small approximation error due to discretization.

3 Feature Extraction

After presenting the various enhancement techniques, which are essential preprocessing steps when dealing with degraded images—caused, for instance, by sensor noise, sweat during scanning, or particularly challenging fingerprints such as those of manual laborers—the next step is the extraction of minutiae. However, before performing this step, it is crucial to obtain a skeletonized or thinned image from a binarized fingerprint.

3.1 Image Binarization

To enable skeletonization, the image must first be binarized. At this stage, the grayscale fingerprint image (with 256 levels of intensity) is converted into a binary image, where black pixels represent the ridges, and white pixels represent the valleys. Various binarization techniques exist; however, we opted for a simple and effective thresholding method.

Each pixel $P(x, y)$ is compared to a predefined threshold M . If the pixel intensity exceeds the threshold, it is set to black (value 1); otherwise, it is set to white (value 0):

$$P_{bin}(x, y) = \begin{cases} 1, & \text{if } P(x, y) > M \\ 0, & \text{otherwise} \end{cases} \quad (40)$$

3.2 Skeletonization of the Image

To facilitate minutiae extraction, the image must be skeletonized. This process involves applying a sequence of morphological erosion operations to reduce the ridge thickness until it is a single pixel wide while maintaining ridge connectivity and the object's overall topology. That is, the continuity of the ridges must be preserved, and no artificial gaps should be introduced.

In our recognition system, we used the skeletonization implementation from `sklearn`, which is based on the Zhang-Suen thinning algorithm [11], wrapped in a function that also removes small objects using `sklearn.morphology`. This algorithm is widely used in fingerprint processing and is known for its reduced computational complexity compared to other thinning algorithms.

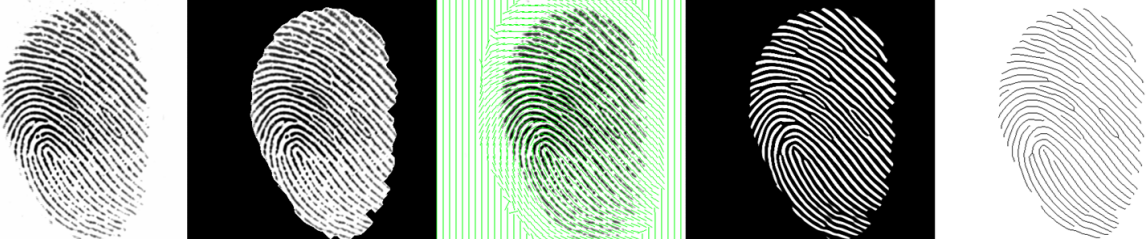


Figure 7: Example of Cappelli's enhancing pipeline until the skeletonize process.

3.3 Minutiae Detection

For minutiae extraction, we adopted the widely used *Crossing Number* (CN) method [12]. This approach is popular due to its simplicity and effectiveness. It uses a skeletonized fingerprint image which is then processed, where white pixels (0) represent the background and black pixels (1) depict the ridges, by analyzing the local neighborhood of each ridge pixel using an 8-neighbor connectivity within a 3×3 window. The CN value for a given ridge pixel P is computed as:

P1	P2	P3
P8	P	P4
P7	P6	P5

$$CN(P) = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i-1}|$$

With: $P_8 = P_0$ $P_i \in \{0,1\}$

where $P_8 = P_0$ ensures a circular index.

Based on the CN value, the classification of minutiae is as follows:

- $CN(P) = 0$: Isolated pixel, generally ignored as it is rare and often results from noise.
- $CN(P) = 1$: Termination candidate.
- $CN(P) = 2$: Normal ridge pixel, not a minutia.
- $CN(P) = 3$: Bifurcation candidate.
- $CN(P) = 4$: Quadruple bifurcation, rare and likely due to noise.

This methodology provides an efficient and accurate extraction of minutiae however, inherent image noise, imperfections in the skeletonization process and poor quality fingerprints, can lead to the detection of minutiae that do not correspond to true ridge discontinuities. These false minutiae, if not removed, can adversely affect matching performance. To tackle this issue, we have implemented an approach to remove them.

3.4 Remove False Minutiae

In order to remove these potential damaging elements, we use two approaches: **Distance-based fuzzy filtering** and **Edge-based filtering**.

The distance-based approach computes the Euclidean distances between all pairs of detected minutiae. By applying a distance threshold, the algorithm identifies minutiae that are too close together, situations that often indicate false positives caused by noise or artifacts from the skeletonization process. Fuzzy rules based on the type of minutiae (e.g., ridge ending vs. bifurcation) are then applied. For instance, if an "ending" and a "bifurcation" minutia are very close, one of them may be removed as they likely represent a single true feature rather than two separate ones. Similarly, if two minutiae of the same type (either both endings or both bifurcations) are too close, they are both flagged for removal. This fuzzy logic approach helps in reducing clusters of spurious minutiae while preserving genuine ones.

The edge-based approach on the other hand addresses the issue of false minutiae that occur near the boundaries of the fingerprint image. It operates by examining the immediate neighborhood of each minutia in the four cardinal directions (east, west, north, and south). If a minutia does not have the expected surrounding valley pixels in these directions, it is likely an artifact produced by edge effects or noise, and therefore it is filtered out.

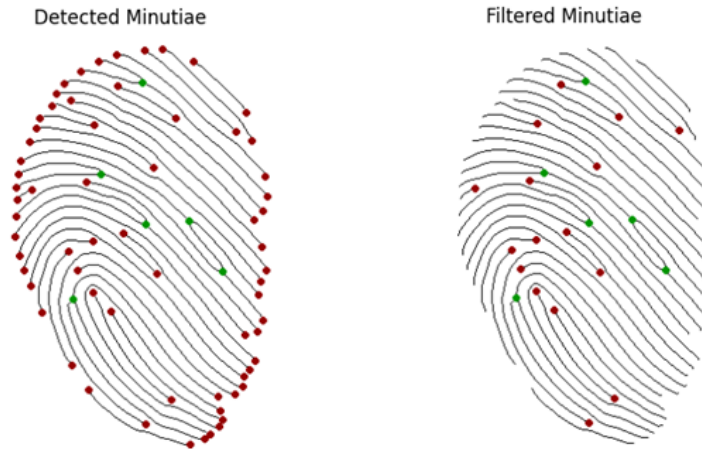


Figure 8: Example of pre and post filtering of false minutiae

4 Minutiae Matching Algorithm

For the matching process, we choose a simple global-structure-based algorithm. Despite the availability of more robust matching strategies, we adhere to the KISS principle in designing our fingerprint recognition system. Even though it is a straightforward method, it is crucial to make it robust; otherwise, impostors could gain access to the system. At the same time, it should not be overly restrictive, to avoid rejecting genuine users.

As we will see in Section 6, the overall performance, particularly when using the Cappelli enhancement pipeline, is quite satisfactory, even under challenging conditions such as massive impostor attempts and highly adversarial datasets like the FVC datasets. Many fingerprints exhibit structural distortions due to the pressure applied on the sensor, while others are partial impressions that were cut off at the edges of the sensor.

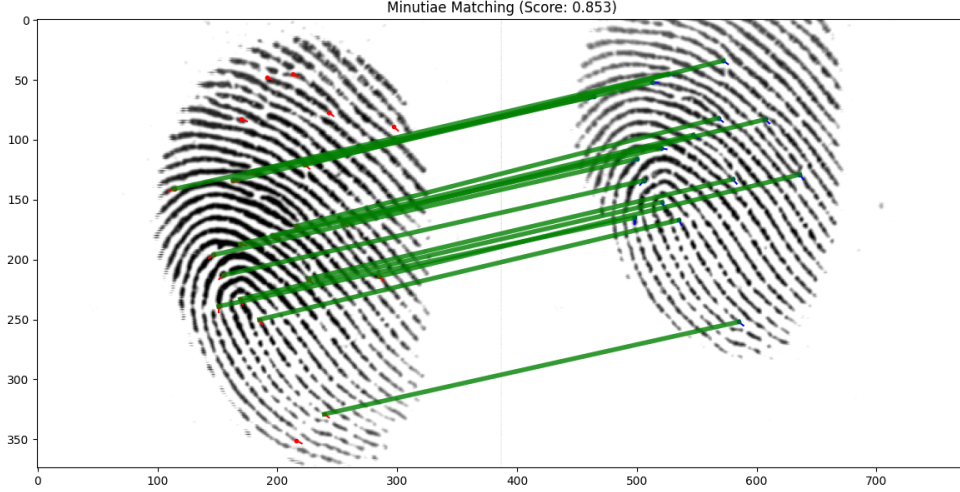


Figure 9: Example of minutiae matching from 101_1.tif and 101_2.tif images from FVC 2002 DB1 B

The implemented minutiae matching algorithm adopts a transformation-invariant approach by aligning two sets of minutiae through translation and rotation correction, ensuring robust matching under different acquisition conditions. This approach is inspired by existing literature, such as [13], which introduces coordinate system-based matching techniques.

4.1 Algorithm Overview

The process begins by iteratively selecting each minutia from both fingerprint templates as a reference. Once a reference minutia is chosen, the coordinate system of the corresponding minutiae set is transformed so that the selected reference minutia is positioned at the origin with its orientation aligned to zero⁴. This step guarantees translation invariance.

To account for rotational variations, the first minutiae set is systematically rotated within a predefined range (e.g., from -5° to 5° in 0.5° increments). For each rotation step, minutiae pairs from both sets are compared based on spatial proximity and orientation similarity. A minutia pair is considered a match if the Euclidean distance d between them is below a predefined threshold $d_{\max}$ and their orientation difference θ does not exceed $\theta_{\max}$. The transformation (i.e., reference minutia and rotation angle) that yields the highest number of matched pairs is selected as the optimal alignment.

4.2 Similarity Score Computation

To quantify the similarity between two minutiae sets, we compute a similarity score S , defined as:

$$S = \sqrt{\frac{M^2}{N_1 N_2}} \quad (41)$$

where M represents the number of matched minutiae pairs, while N_1 and N_2 denote the total minutiae count in the two sets. This formulation ensures that the score remains invariant to the size of the fingerprint templates and emphasizes the proportion of matched minutiae.

4.3 Key Parameters

The accuracy and robustness of the matching process depend on several key parameters:

- **Maximum Distance ($d_{\max}$):** The maximum Euclidean distance allowed between two minutiae for them to be considered a match. A larger value increases tolerance to minor distortions but may lead to false matches.
- **Maximum Angular Difference ($\theta_{\max}$):** The maximum allowable orientation difference between two minutiae. This parameter controls sensitivity to rotational variations and should be tuned according to the acquisition conditions.

⁴The orientation of a minutia is estimated using the orientation map of the fingerprint image.

- **Rotation Range and Step:** The extent and granularity of the rotation applied to one minutiae set. A finer step size improves precision but increases computational complexity.

By carefully tuning these parameters, the algorithm achieves a balance between accuracy and efficiency, making it suitable for fingerprint recognition tasks in both biometric identification and verification.

5 Singular Points in Fingerprints

Singular points are distinctive locations in a fingerprint where the ridge orientation exhibits discontinuities. The three primary types of singular points are the *loop*, the *delta*, and the *whorl*. The loop is typically the uppermost point of the innermost ridge, while the delta is the location where three ridge flows converge, forming a triangular pattern. A whorl, on the other hand, is a circular or spiral pattern that often contains two loops.

To detect singular points, the *Poincaré Index* method [12] is widely used. This method computes the sum of orientation changes along a closed path surrounding a given pixel in the orientation field. If the computed index is approximately $+\pi$ (or $+\frac{1}{2}$ in a normalized form), the point is classified as a loop. If the index is around $-\pi$ (or $-\frac{1}{2}$), the point is a delta. A value near $+2\pi$ suggests the presence of a whorl, but for the most orientation maps these singularities are chosen as two loops near each other. This simplification is due to the difficulty of obtaining a precise orientation estimate in regions with complex circular structures. A value close to zero indicates a non-singular region.

Mathematically, for a discrete orientation map, the Poincaré Index at a point (x, y) is computed by summing the angular differences between adjacent pixels along an 8-connected neighborhood. The formula is:

$$PI(x, y) = \frac{1}{2\pi} \sum_{i=0}^{N-1} \Delta\theta_i \quad (42)$$

where $\Delta\theta_i$ is the angle difference between successive orientation values along the closed path.

The accuracy of singular point detection depends on the quality of the orientation field and preprocessing techniques like Gaussian smoothing. Moreover, thresholding is applied to ensure robustness against noise in real-world fingerprint images. Since whorls often consist of multiple loops, their detection requires analyzing regions with multiple high-magnitude Poincaré Index values.

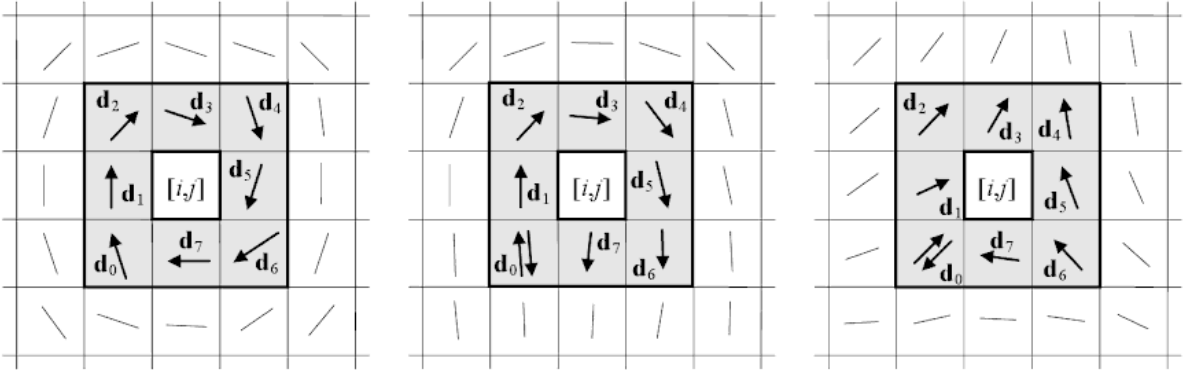


Figure 10: Examples of Poincaré index computation in the 8-neighborhood of points belonging (from left to right) to a whorl, loop, and delta singularity, respectively.

5.1 Role of Singularities in Matching

Singular points can act as **discriminative features** in our fingerprint matching process. If two fingerprints exhibit different singularities, they may be considered non-comparable. However, this approach assumes high reliability in singularity detection, which depends on the accuracy of the orientation field estimation during the enhancement phase. Additionally, we must ensure that the fingerprint is not cropped by the sensor edges, as this could lead to missing or misidentified singular points, compromising the matching process.

To enhance robustness, an alternative approach is introduced a **gain mechanisms**. Instead of discarding comparisons between fingerprints with different singularities, a gain factor is applied to the matching score if during the global

minutiae structure matching, singularities match. This method reduces the risk of false acceptances caused by only minutiae matching analysis.

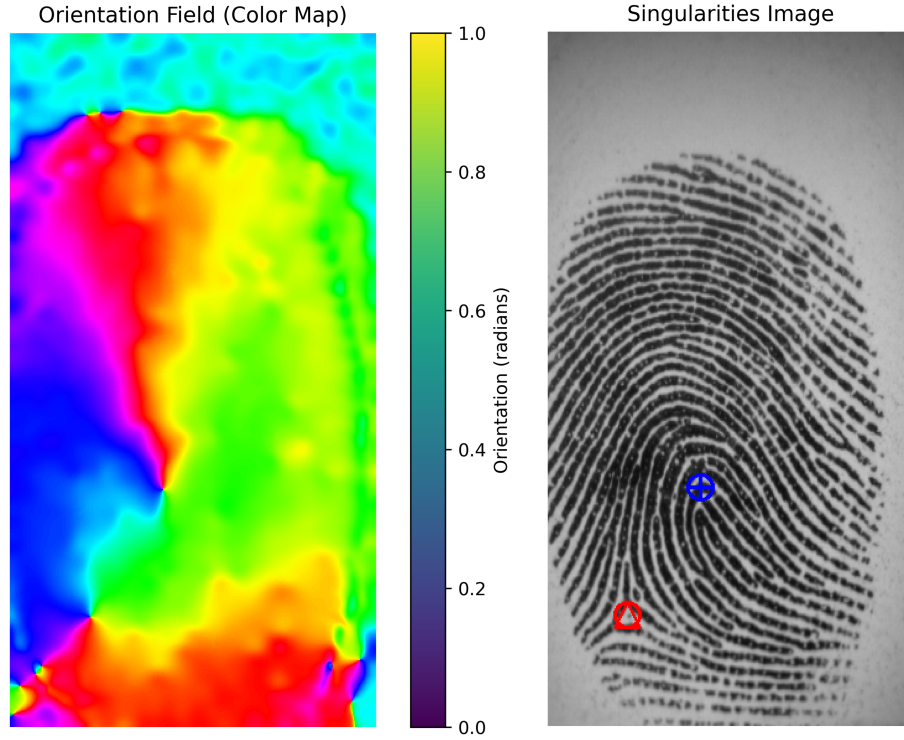


Figure 11: Example of singularity extraction output. The left image represents the orientation map computed using GBFOE, where the peaks (visible as spiral-like patterns) correspond to the singularities. The right image shows the output of the *Poincaré Index* method, where the red triangle represents the delta and the blue circle denotes the loop.

6 Evaluation

The objective of this project was to compare various fingerprint datasets and enhancement approaches in order to evaluate our system's performance. For a comprehensive assessment, we computed several evaluation metrics, including the decidability factor on processed fingerprints, SRR (System response reliability), ROC (Receiver Operating Characteristic) and direct comparisons of the EER (Equal Error Rate) under different gallery and impostor ratios. Additionally, we collected detection rank data for closed-set identification scenarios. In particular we will set up comparisons between different FVC2002 datasets which deploy different sensors for fingerprint capture. We decided to use:

- **DB1:** optical sensor "TouchView II" by Identix
- **DB2:** optical sensor "FX2000" by Biometrika
- **DB3:** capacitive sensor "100 SC" by Precise Biometrics
- **DB4:** synthetic fingerprint generation
- **Neurotechnology CrossMatch:** optical sensor "Verifier 300" by Cross Match Technologies

6.1 Decidability

Decidability is a measure of the ability of a biometric system to distinguish between samples belonging to the same identity (intra-class) and samples belonging to different identities (inter-class). In our study, decidability is computed as the absolute difference between the average intra-class similarity and the average inter-class similarity, normalized by the spread of the similarity distributions. Mathematically, it is defined as:

$$decidability = \frac{|\bar{s}_{intra} - \bar{s}_{inter}|}{\sqrt{0.5 * \sigma_{intra}^2 + \sigma_{inter}^2}} \quad (43)$$

where:

- $\bar{s}_{intra}$ is the average intra-class similarity score
- $\bar{s}_{inter}$ is the average inter-class similarity score
- σ_{intra} is the standard deviation for the intra-class similarities
- σ_{inter} is the standard deviation for the inter-class similarities

To compute these statistics, we first generate a complete ALL-against-ALL score matrix for our dataset by comparing every template with every other template. This comprehensive matrix allows us to derive robust estimates of both the central tendency and the dispersion of intra-class and inter-class similarity scores.

This formula normalizes the difference between the mean intra-class and inter-class similarities by a factor that accounts for the spread (variance) of these scores.

Dataset	Avg Decidability	Avg Intra-class	Avg Inter-class	Avg std dev Intra-class	Avg std dev Inter-class
FVC DB1	4.210	0.662	0.289	0.110	0.055
FVC DB2	6.018	0.702	0.267	0.089	0.043
FVC DB3	3.445	0.640	0.349	0.089	0.058
FVC DB4	3.468	0.640	0.321	0.111	0.057
NC	9.114	0.833	0.297	0.048	0.048
Overall	5.251	0.695	0.304	0.089	0.052

Table 1: Average Decidability and Class Similarity Metrics - Raffaele Cappelli enhancement

Dataset	Avg Decidability	Avg Intra-class	Avg Inter-class	Avg std dev Intra-class	Avg std dev Inter-class
FVC DB1	3.403	0.598	0.290	0.118	0.051
FVC DB2	4.316	0.587	0.256	0.097	0.039
FVC DB3	2.680	0.583	0.350	0.097	0.053
FVC DB4	3.370	0.597	0.305	0.105	0.051
NC	7.875	0.775	0.298	0.055	0.046
Overall	4.328	0.628	0.299	0.094	0.048

Table 2: Average Decidability and Class Similarity Metrics - Peter Kovesi enhancement

These tables consolidate five key parameters that characterize the overall decidability of the biometric system. We decided to track the average decidability, the average intra-class similarity, the average inter-class similarity, the average standard deviation of the intra-class similarities and lastly the average standard deviation of inter-class similarities. Notably, using Raffaele Cappelli’s enhancement process, the Neurotechnology CrossMatch (NC) dataset exhibits the highest decidability, while FVC2002 Dataset 3 shows the lowest decidability. This trend is consistent regardless of the enhancement pipeline applied.

6.2 SRR (System response reliability)

In order to gain an insight on the actual correctness of the system's decision on whether a corresponding fingerprint is found, we compute the System Response Reliability. The System Response Reliability (SRR) is an index that measures the reliability of a single response from an identification biometric system, specifically its ability to distinguish between genuine subjects and impostors for a given probe. Unlike overall system performance metrics as ROC, which is calculated over a set of trials and across different thresholds, SRR aims to provide a measure of confidence in the outcome of a single identification attempt.

The core concept of SRR is that it assesses the degree of "confusion" among potential matches for a given probe during identification. The underlying idea is that if the top-ranked match is significantly different from other candidates in the gallery, the system's response is considered more reliable. Conversely, if several gallery members have similar scores to the top match, the response is deemed less reliable due to potential ambiguity.

In our system, the System Response Rate (SRR) is computed from a reliability measure, denoted as the ϕ_1 score, which represents the relative gap between the highest and second-highest similarity scores for a given sample. Specifically, the ϕ_1 score is calculated as:

$$\phi_1 = \frac{(1 - \text{second_best_similarity}) - (1 - \text{best_similarity})}{(1 - \text{worst_similarity})} \quad (44)$$

assuming the best similarity is nonzero. This formulation captures how distinctly the best match stands out relative to the next candidate. The SRR is then derived by comparing each ϕ_1 score to an optimal threshold (T) that is determined based on the trade-off between genuine and impostor identification errors. The SRR is computed as:

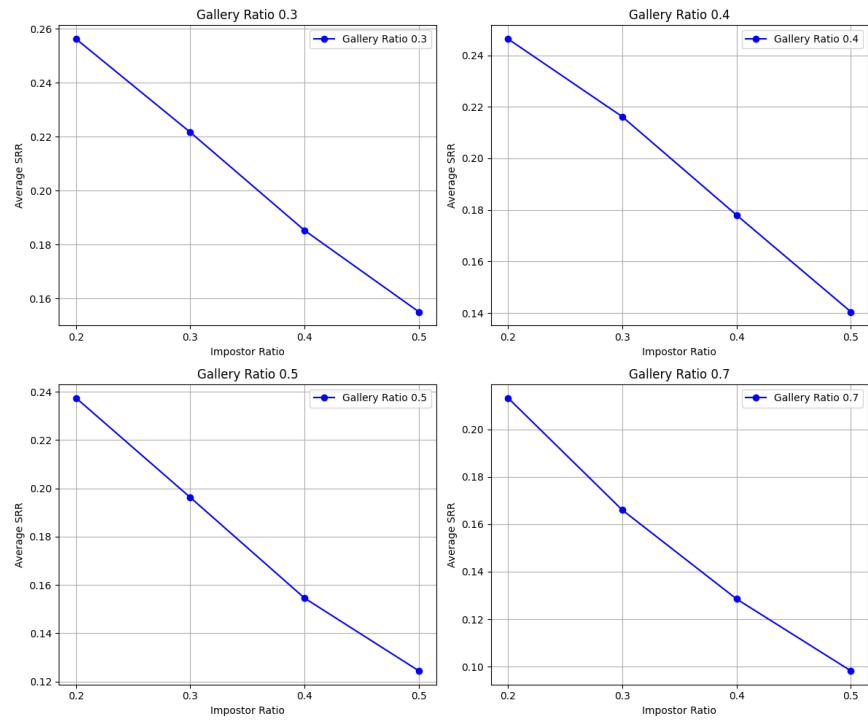
$$SRR = \begin{cases} \frac{\phi_1 - T}{1 - T}, & \text{for } \phi_1 > T \\ \frac{T - \phi_1}{T}, & \text{for } \phi_1 \leq T \end{cases}$$

This two-branch normalization ensures that:

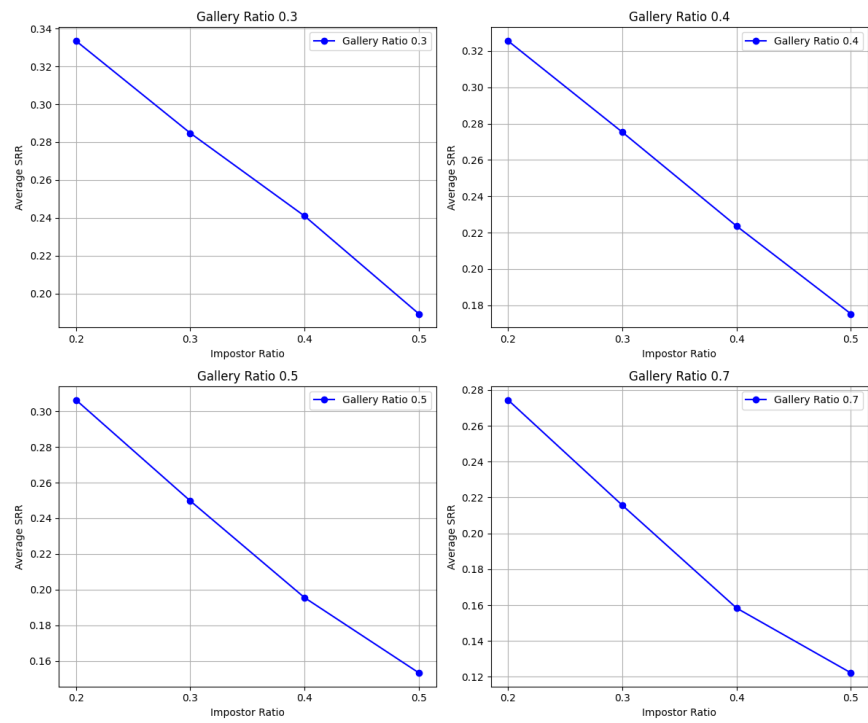
- A positive SRR indicates that the ϕ_1 score exceeds the threshold, suggesting a reliable, genuine match.
- A ϕ_1 score that falls below the threshold will be set to 0, signaling potential impostor identification.

In essence, SRR provides a normalized measure of the system's confidence in its identification decision. It reflects the margin by which a sample's reliability measure surpasses or falls short of the optimal decision boundary. This makes the SRR a valuable metric for comparing performance across different system configurations or datasets in biometric identification contexts.

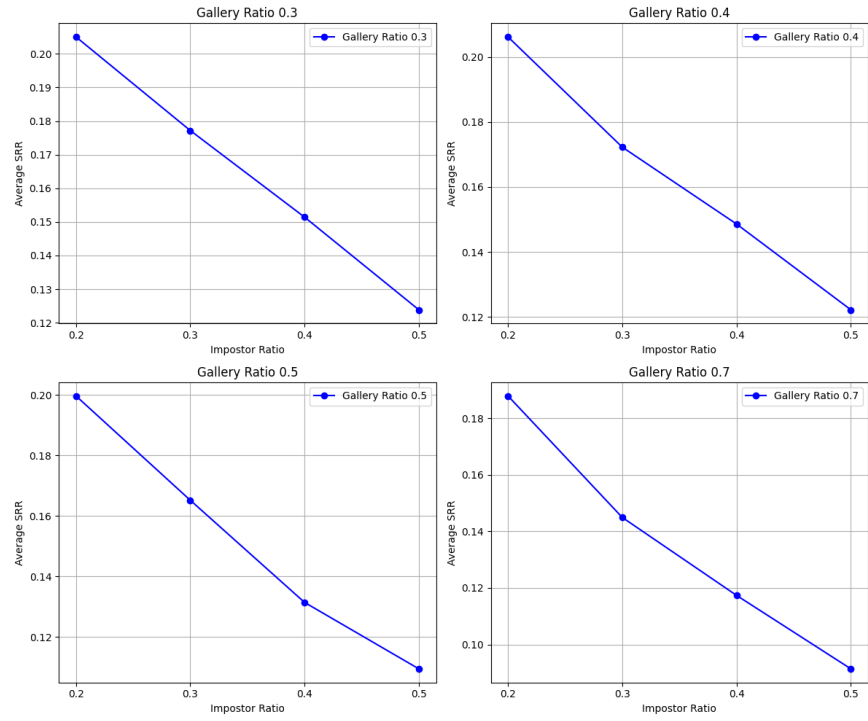
6.2.1 SRR for FVC2002 Dataset 1 - Raffaele Cappelli Enhancement



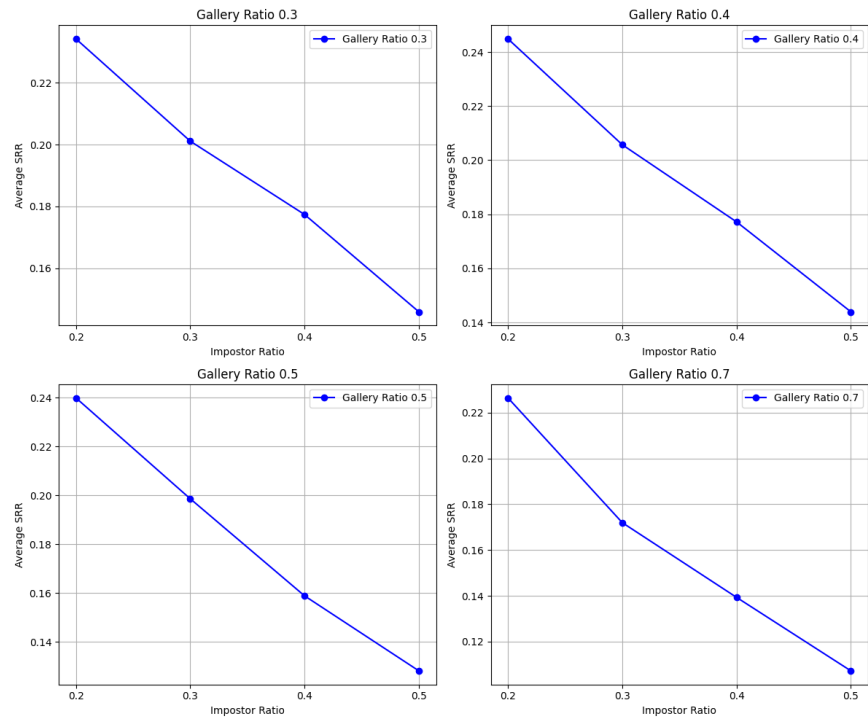
6.2.2 SRR for FVC2002 Dataset 2 - Raffaele Cappelli Enhancement



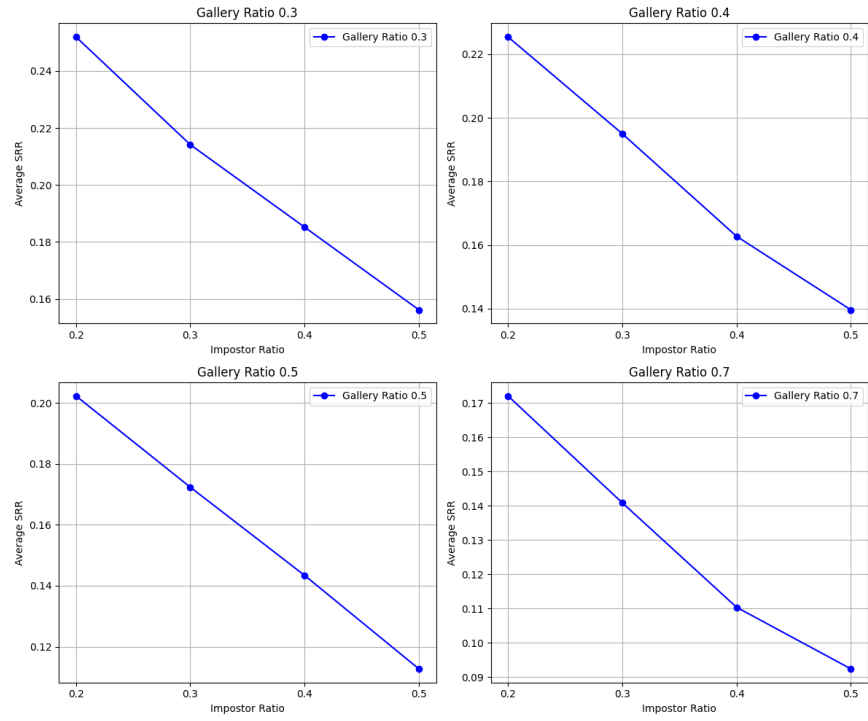
6.2.3 SRR for FVC2002 Dataset 3 - Raffaele Cappelli Enhancement



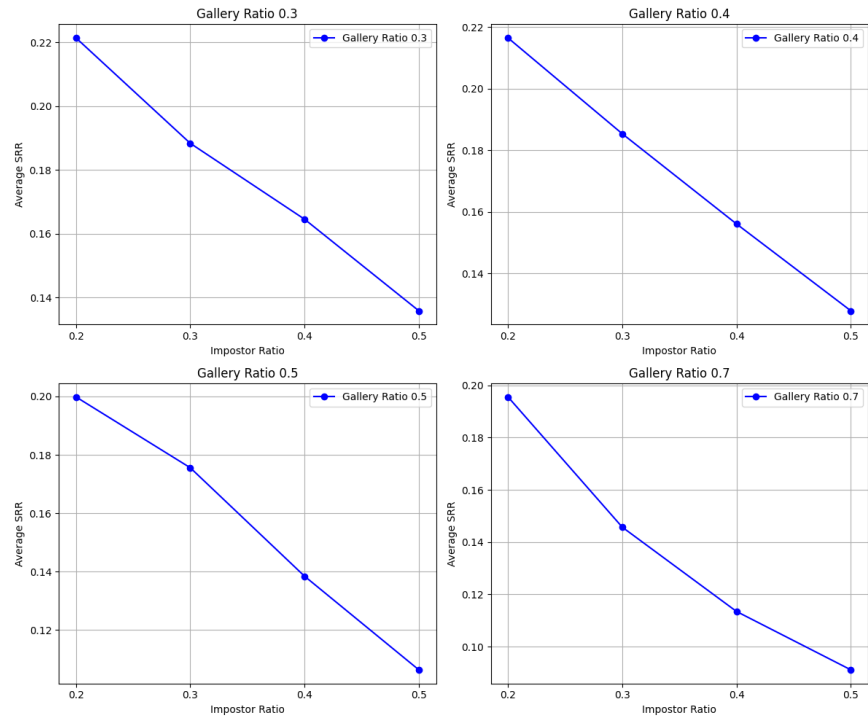
6.2.4 SRR for FVC2002 Dataset 4 - Raffaele Cappelli Enhancement



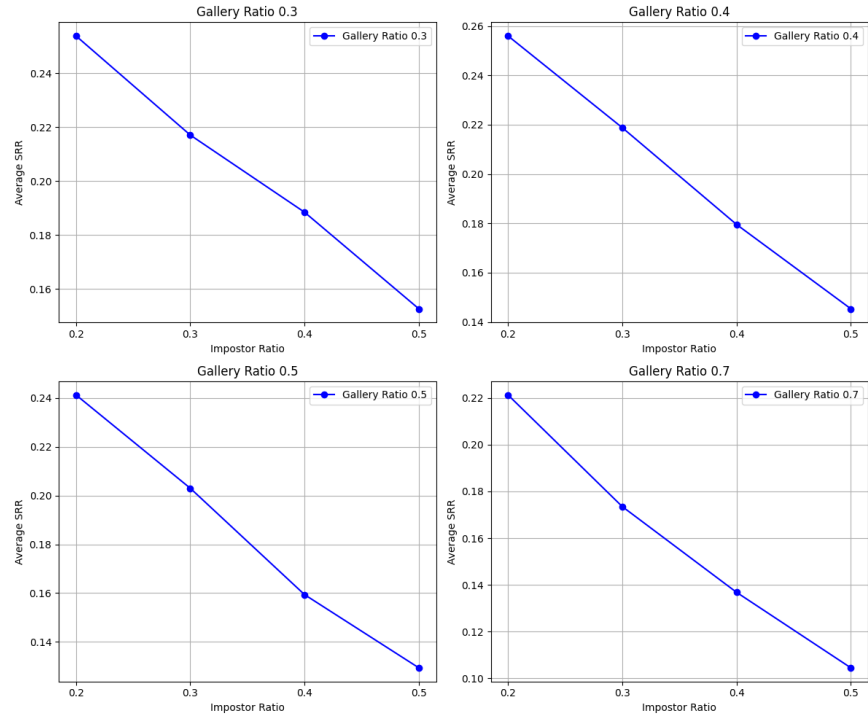
6.2.5 SRR for Neurotechnology CrossMatch Dataset - Raffaele Cappelli Enhancement



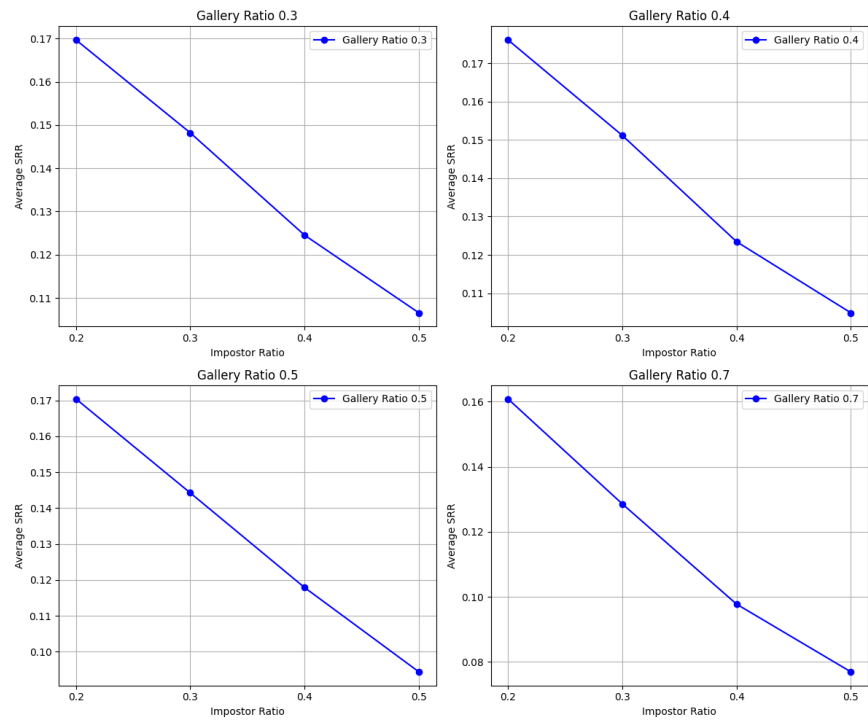
6.2.6 SRR for FVC2002 Dataset 1 - Peter Kovesi Enhancement



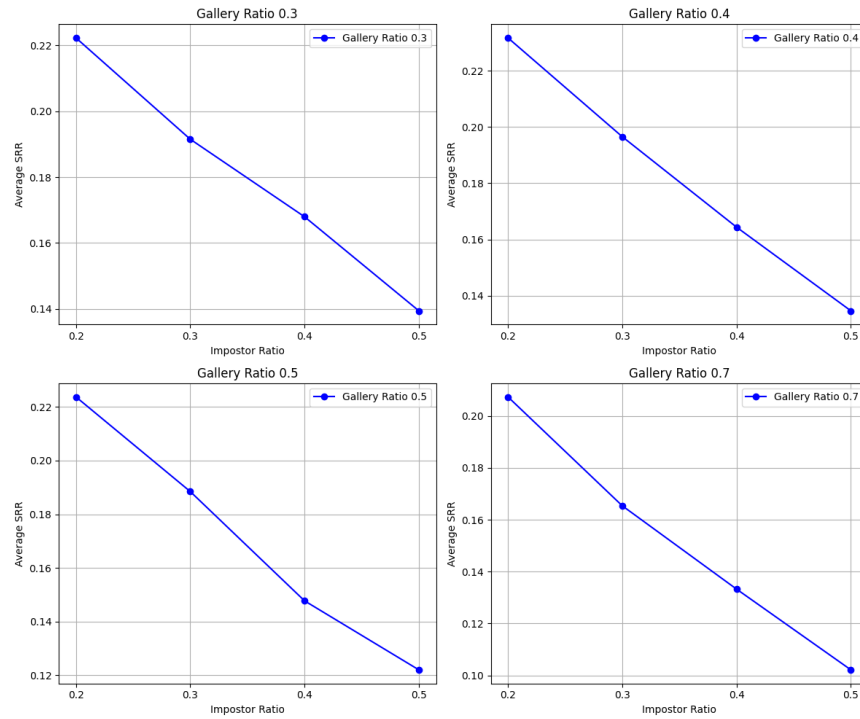
6.2.7 SRR for FVC2002 Dataset 2 - Peter Kovesi Enhancement



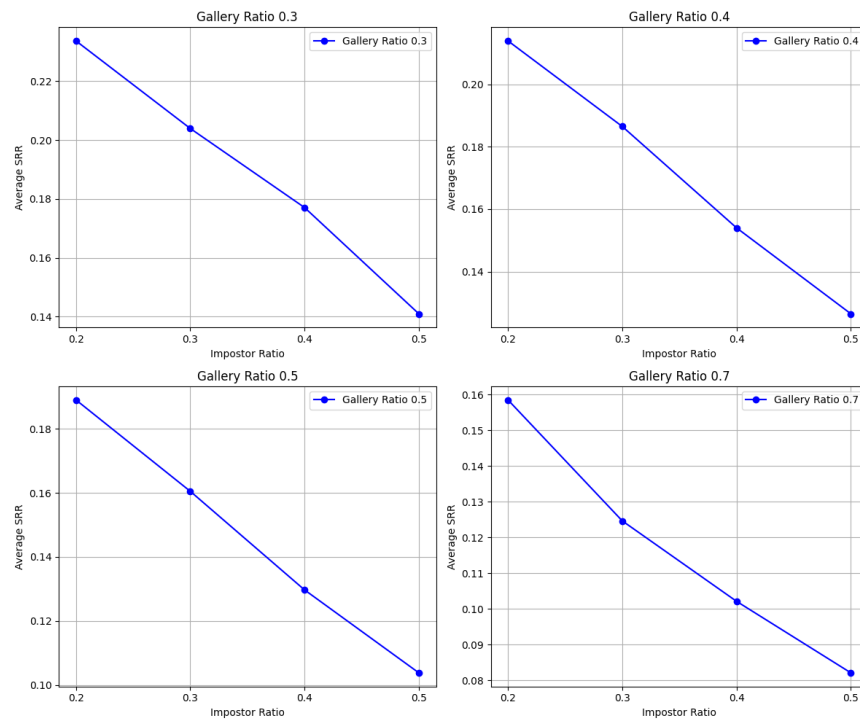
6.2.8 SRR for FVC2002 Dataset 3 - Peter Kovesi Enhancement



6.2.9 SRR for FVC2002 Dataset 4 - Peter Kovesi Enhancement



6.2.10 SRR for Neurotechnology Crossmatch - Peter Kovesi Enhancement



Dataset	Average SRR loss
FVC DB1	0.11
FVC DB2	0.14
FVC DB3	0.08
FVC DB4	0.10
NC	0.09
Overall	0.10

Table 3: Average SRR loss metrics - Raffaele Cappelli enhancement

Dataset	Average SRR loss
FVC DB1	0.11
FVC DB2	0.10
FVC DB3	0.07
FVC DB4	0.09
NC	0.08
Overall	0.09

Table 4: Average SRR loss metrics - Peter Kovesi enhancement

From these graphs we can see how different gallery and impostor ratios go and impact the overall performance of the system and we can line out a common trend present across all datasets and present on both enhancement processes. We notice that as the gallery ratio is changed along with the impostor ratio, the SRR steadily decreases. This phenomenon is due to the fact that as the gallery set increases, the number of templates of each genuine identity increases. As the impostor ratio increases, more overall identities are included in the impostor set. This leads to a greater chance for an impostor to potentially match with one of the gallery templates, which leads to an overall lower SSR. The idea behind the SRR is that the higher it is, the greater amount of genuine users it was able to identify.

From these graphs, we observe that varying the gallery and impostor ratios significantly affects the overall performance of the system, with a clear trend evident across all datasets and both enhancement processes. Specifically, as the gallery ratio increases and the impostor ratio becomes higher, the System Response Rate (SRR) steadily decreases.

This trend can be explained by the fact that a higher gallery ratio results in a larger number of templates for each genuine identity. While this provides more data for matching, it also increases the likelihood of slight variations between different genuine samples, potentially reducing the discriminative margin between the true match and other close matches. Moreover as more identities are added to the impostor set, the probability that an impostor template closely resembles one of the gallery templates increases. This raises the chance of false matches, thereby lowering the SRR.

The SRR is designed to quantify the system's identification confidence. A higher SRR indicates that the system has a larger margin by which genuine users are correctly identified relative to impostors. Thus, a decreasing SRR with increasing gallery and impostor ratios suggests that the system's ability to distinguish genuine matches from impostor matches diminishes when the number of potential mismatches increases.

In summary, these results highlight the importance of carefully selecting the gallery and impostor ratios to optimize the balance between identification accuracy and the system's overall robustness against false matches.

6.3 ROC (Receiver Operating Characteristic)

The Receiver Operating Characteristic (ROC) curve is a fundamental tool used to evaluate the performance of biometric systems, particularly in verification mode. It graphically illustrates the trade-off between the False Accept Rate (FAR) and the Genuine Accept Rate (GAR), which is equivalent to 1 minus the False Rejection Rate (FRR), as the system's decision threshold is varied.

Furthermore ROC curves are used to report on the detection performance for verification tasks and to summarize the performance of a biometric system at different operating points. They show how well a biometric system can detect a genuine user while avoiding the acceptance of impostors.

To assess the overall performance, we pooled all the ϕ_1 scores (used to compute the System Response Rate, SRR) across various gallery and impostor ratio configurations and plotted an aggregated ROC curve for each dataset. The Area Under the Curve (AUC) was then computed as a summary metric, with higher AUC values indicating better discriminative performance.

Dataset	Avg AUC across all curves
FVC DB1	0.994
FVC DB2	0.991
FVC DB3	0.964
FVC DB4	0.978
NC	1.000
Overall	0.985

Table 5: Average ROC Metrics - Raffaele Cappelli enhancement

Dataset	Avg AUC across all curves
FVC DB1	0.983
FVC DB2	0.979
FVC DB3	0.951
FVC DB4	0.951
NC	1.000
Overall	0.972

Table 6: Average ROC Metrics - Peter Kovesi enhancement

From these tables, we can determine the overall ROC performance across all datasets and observe the impact of different enhancement pipelines on the system's performance. Notably, the Neurotechnology CrossMatch (NC) dataset achieved the highest AUC values under both enhancement processes, indicating excellent discriminative ability, whereas FVC2002 Dataset 3 consistently demonstrated the lowest overall performance.

This study tells us that as the gallery and impostor ratios vary, the system's performance, measured by the ROC and AUC, is impacted. The observed trends are consistent across different datasets and enhancement pipelines.

6.4 EER comparisons

EER is the operating point where the False Accept Rate (FAR) equals the False Rejection Rate (FRR) and serves as a useful metric for system performance where the lower the EER, the better the discriminative capability of the biometric system. By varying the gallery and impostor ratios, we can examine how changes in enrollment size and impostor population affect the system's performance. This analysis provides critical insights into the trade-offs inherent in biometric recognition systems and aids in optimizing their overall configuration.

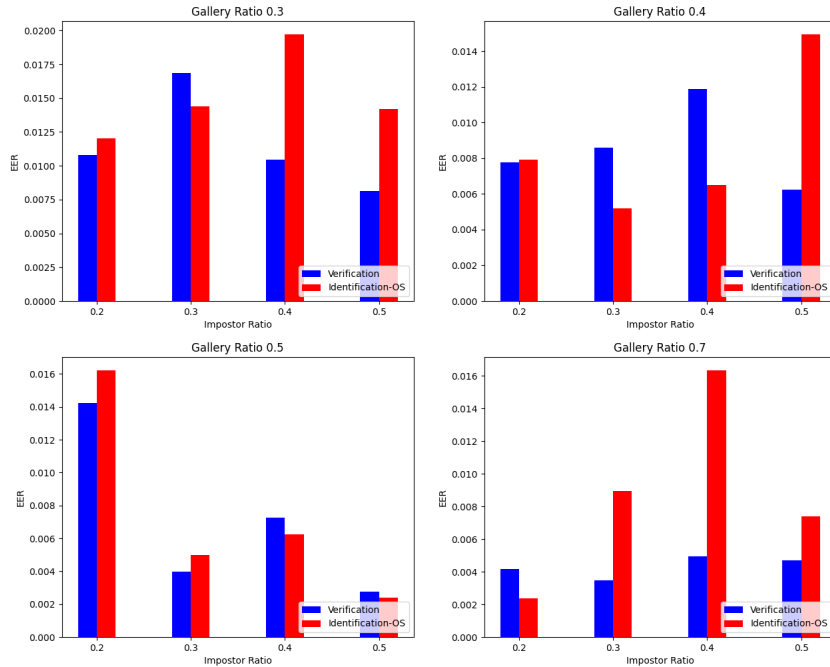
To assess the overall performance of our system across various scenarios, we compute Equal Error Rate (EER) comparisons based on different gallery and impostor ratios. The gallery ratio defines the proportion of probe samples from our datasets that are included in the gallery, while the impostor ratio specifies the percentage of identities used as impostors. In our study, we evaluate a range of scenarios by employing gallery ratios from 30% to 70% and impostor ratios from 20% to 50%.

Moreover the comparisons of the EER values have been done by directly comparing the EER values obtained in verification mode and in identification open set.

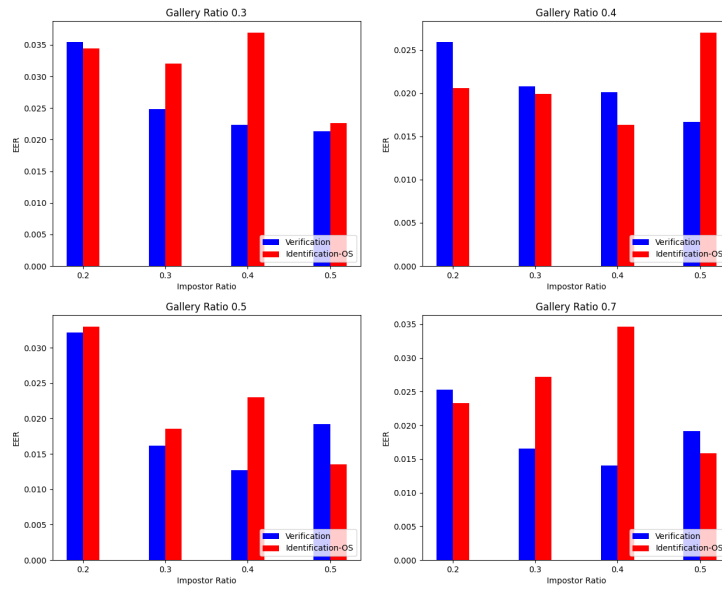
In **verification mode**, the user initially performs an identity claim. The biometric system will then perform a one-to-one comparison between the probe template and a stored template or stored templates of the claim identity. The system then computes a similarity score and makes a decision on whether to accept or reject that user. It will be able to make this decision by checking whether the similarity score is above or below a certain threshold.

In contrast, in an **identification open-set** scenario, the user does not perform an identity claim and it is up to the system to perform a one-to-many search against all gallery templates. In this phase the system needs to identify the most likely match for the user's identity and at the same time decide on whether the input sample belongs to any of the enrolled identities.

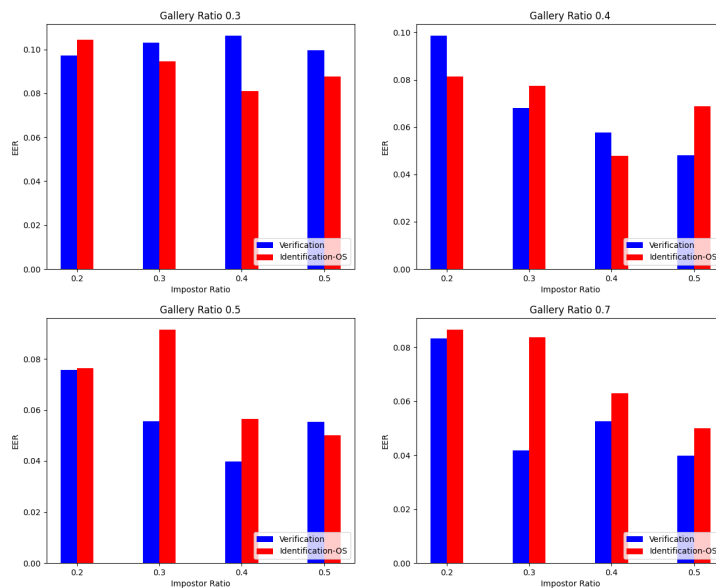
6.4.1 EERs for FVC2002 Dataset 1 - Raffaele Cappelli Enhancement



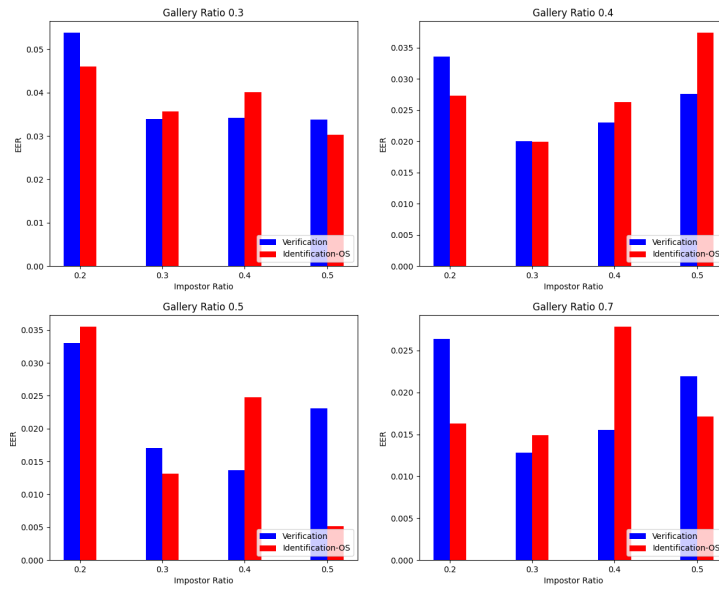
6.4.2 EERs for FVC2002 Dataset 2 - Raffaele Cappelli Enhancement



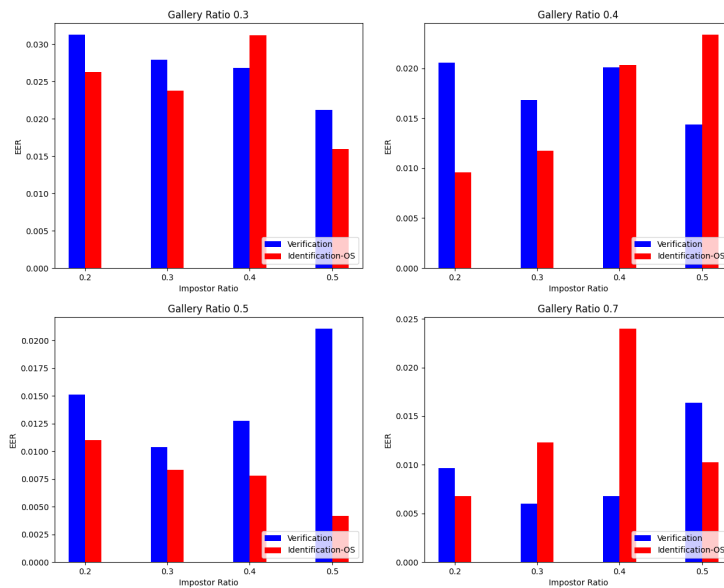
6.4.3 EERs for FVC2002 Dataset 3 - Raffaele Cappelli Enhancement



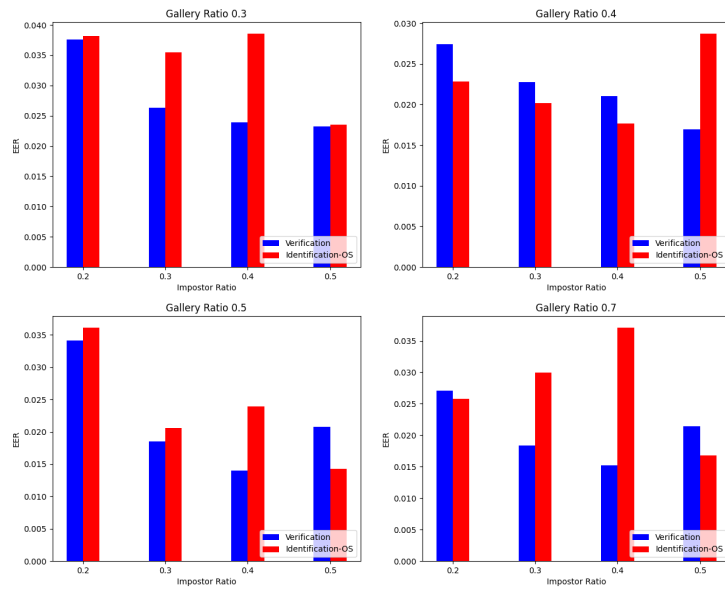
6.4.4 EERs for FVC2002 Dataset 4 - Raffaele Cappelli Enhancement



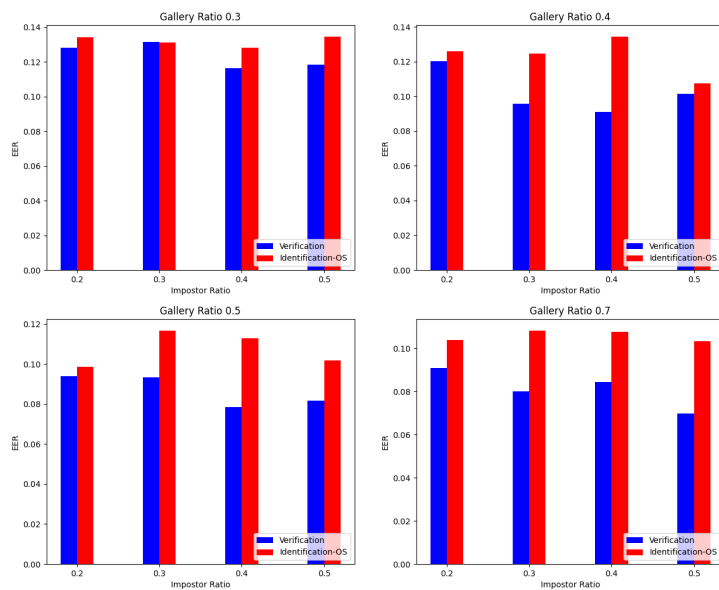
6.4.5 EERs for FVC2002 Dataset 1 - Peter Kovesi Enhancement



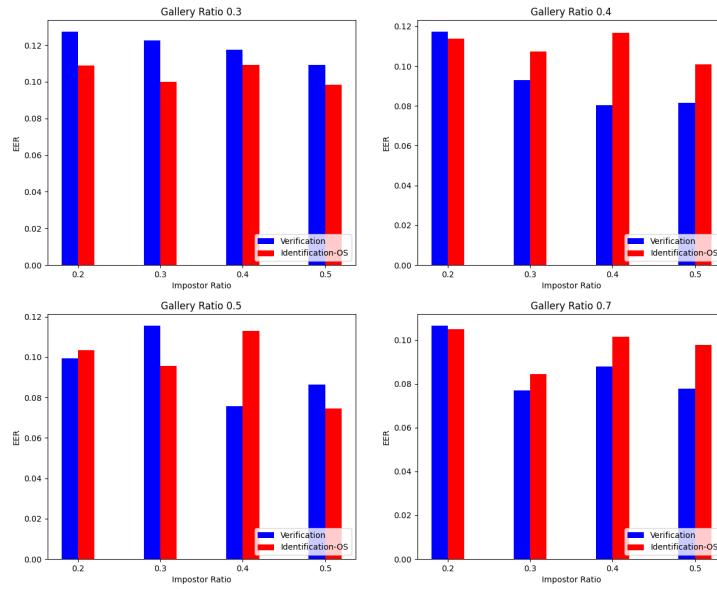
6.4.6 EERs for FVC2002 Dataset 2 - Peter Kovesi Enhancement



6.4.7 EERs for FVC2002 Dataset 3 - Peter Kovesi Enhancement



6.4.8 EERs for FVC2002 Dataset 4 - Peter Kovesi Enhancement



Dataset	Avg EER for Verification	Avg EER for Identification (open-set)
FVC DB1	0.018	0.017
FVC DB2	0.033	0.034
FVC DB3	0.075	0.076
FVC DB4	0.041	0.040
NC	0	0
Overall	0.033	0.033

Table 7: Average EER Metrics - Raffaele Cappelli enhancement

Dataset	Avg EER for Verification	Avg EER for Identification (open-set)
FVC DB1	0.032	0.033
FVC DB2	0.032	0.033
FVC DB3	0.109	0.111
FVC DB4	0.100	0.099
NC	0	0
Overall	0.054	0.055

Table 8: Average EER Metrics - Peter Kovesi enhancement

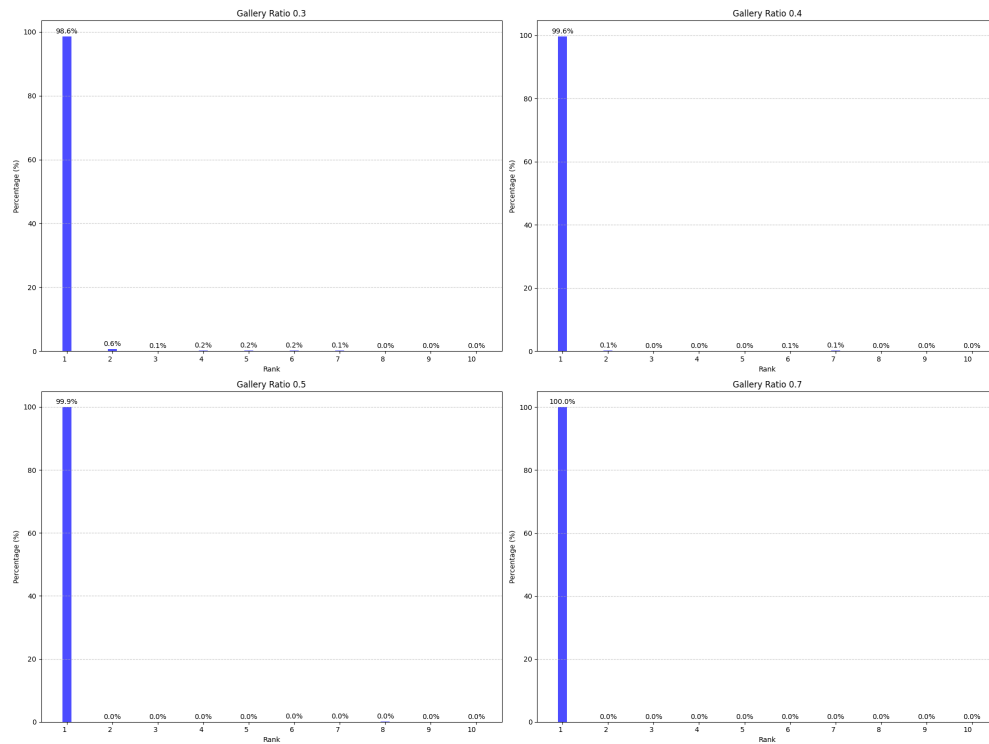
7 Ranks for Identification in closed set

In a closed-set identification scenario, it is assumed that the probe identity is guaranteed to be present within the gallery. Consequently, the system operates without the need for impostor comparisons or threshold determination. Instead, upon presentation of a probe, the system generates a ranked list of potential identities.

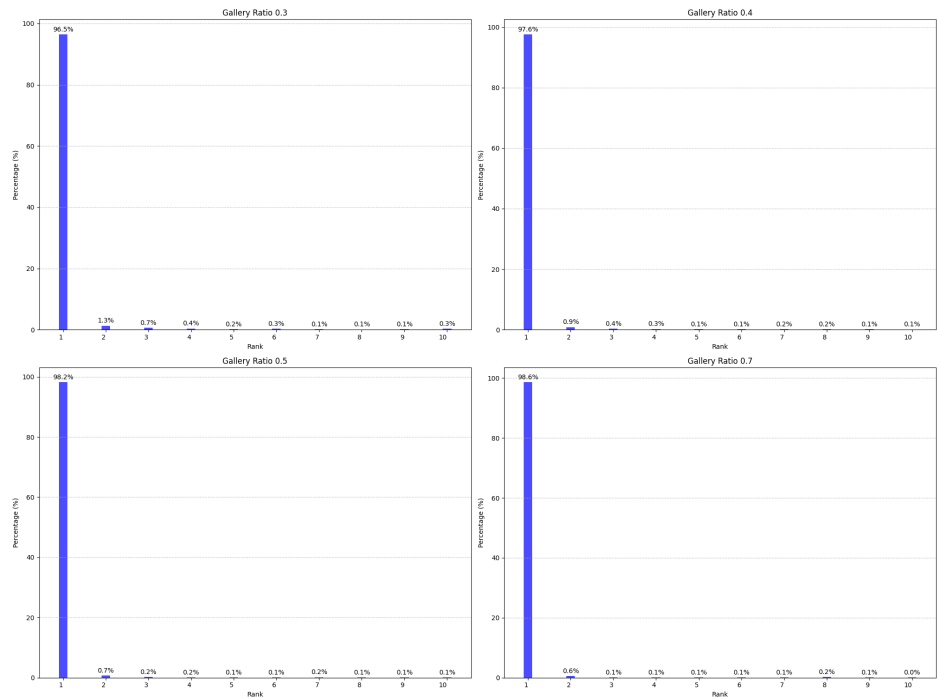
Given that each identity in the gallery is represented by multiple instances, the ranking process leverages the concept of optimal intra-class comparisons. Specifically, for each gallery subject, the system identifies the highest similarity score achieved between the probe and any of the subject's gallery instances. The ranked list is then constructed based on these maximum intra-class similarity scores.

This approach ensures that the ranking reflects the best possible match for each gallery identity, thereby maximizing the likelihood of accurate identification within the closed set.

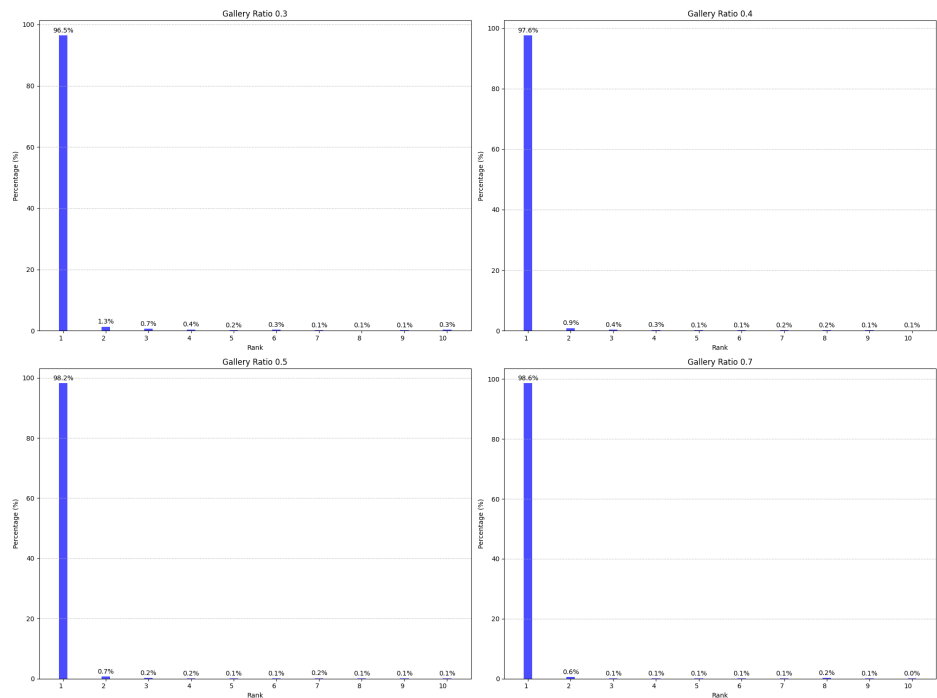
7.0.1 Ranks for FVC2002 Dataset 1 - Raffaele Cappelli Enhancement



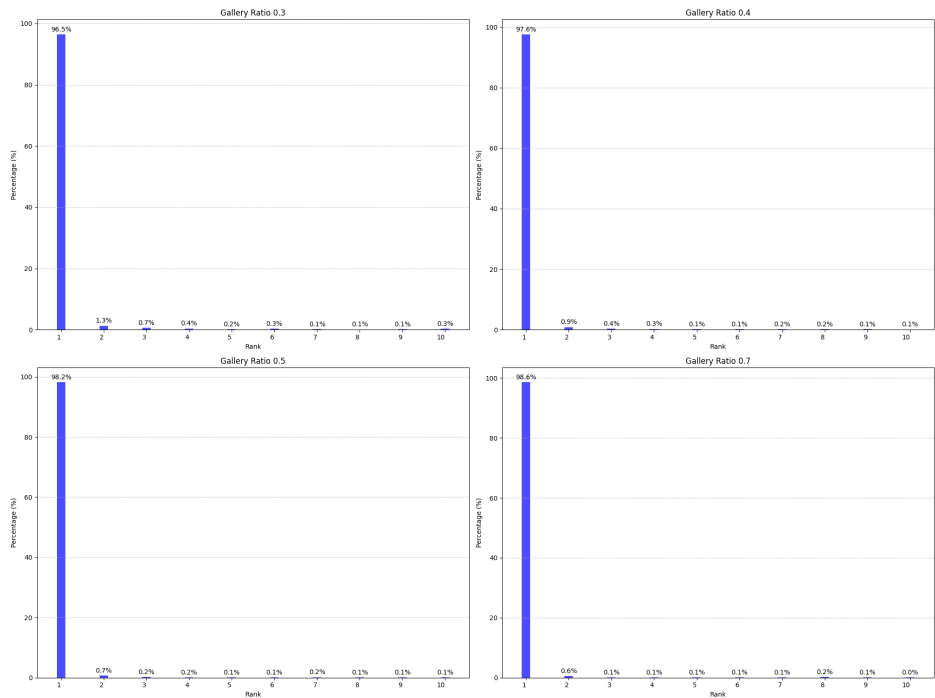
7.0.2 Ranks for FVC2002 Dataset 2 - Raffaele Cappelli Enhancement



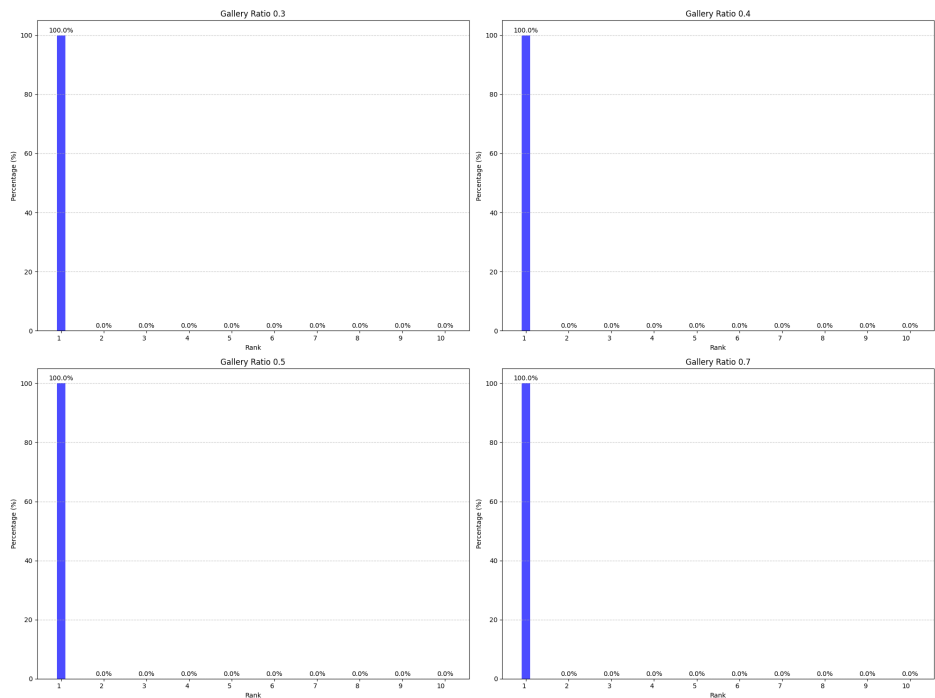
7.0.3 Ranks for FVC2002 Dataset 3 - Raffaele Cappelli Enhancement



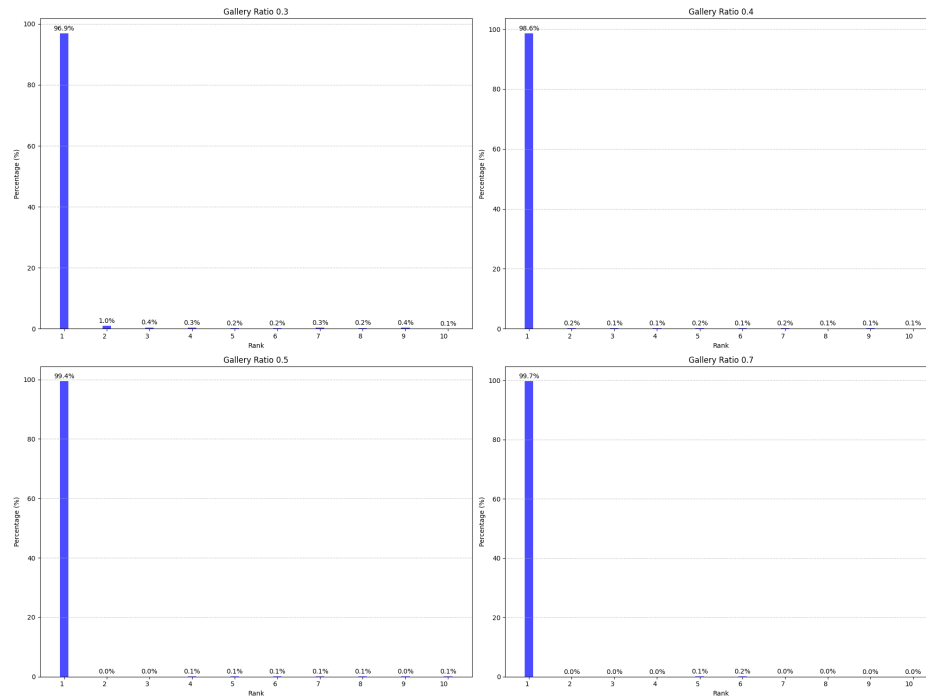
7.0.4 Ranks for FVC2002 Dataset 4 - Raffaele Cappelli Enhancement



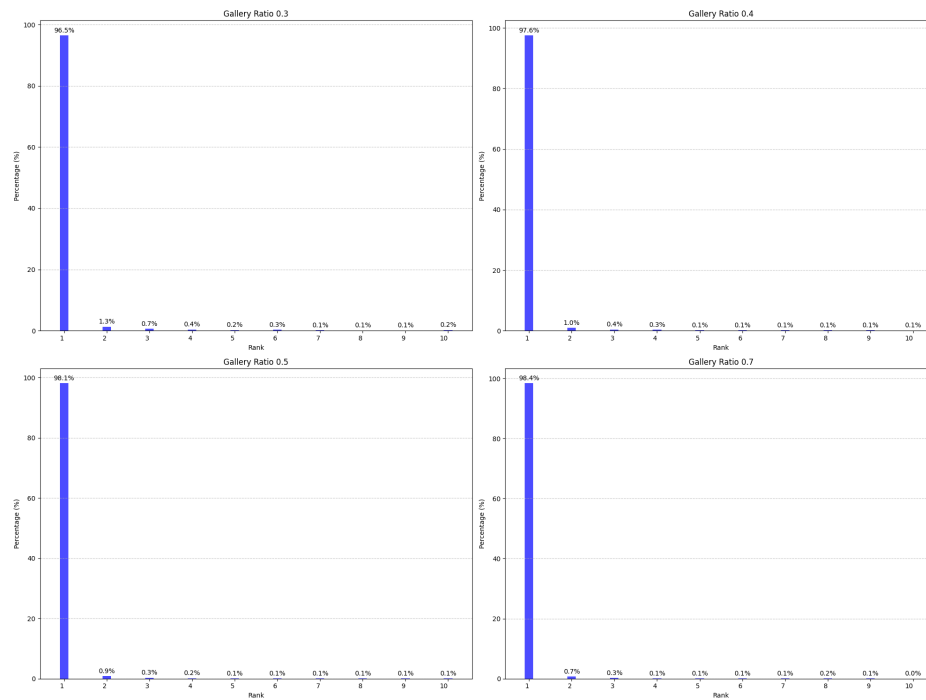
7.0.5 Ranks for Neurotechnology CrossMatch - Raffaele Cappelli Enhancement



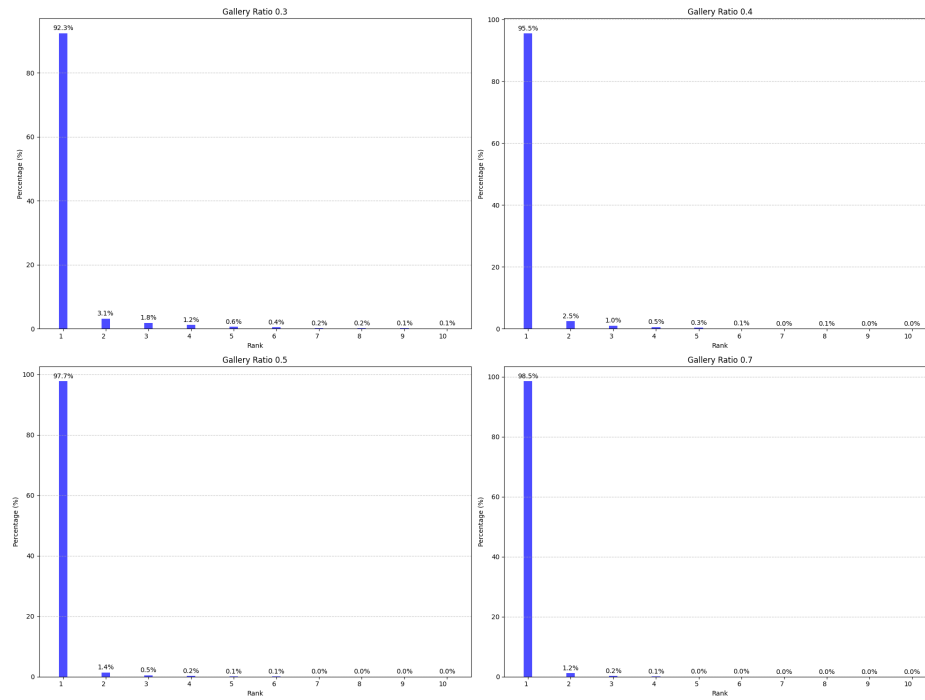
7.0.6 Ranks for FVC2002 Dataset 1 - Peter Kovesi Enhancement



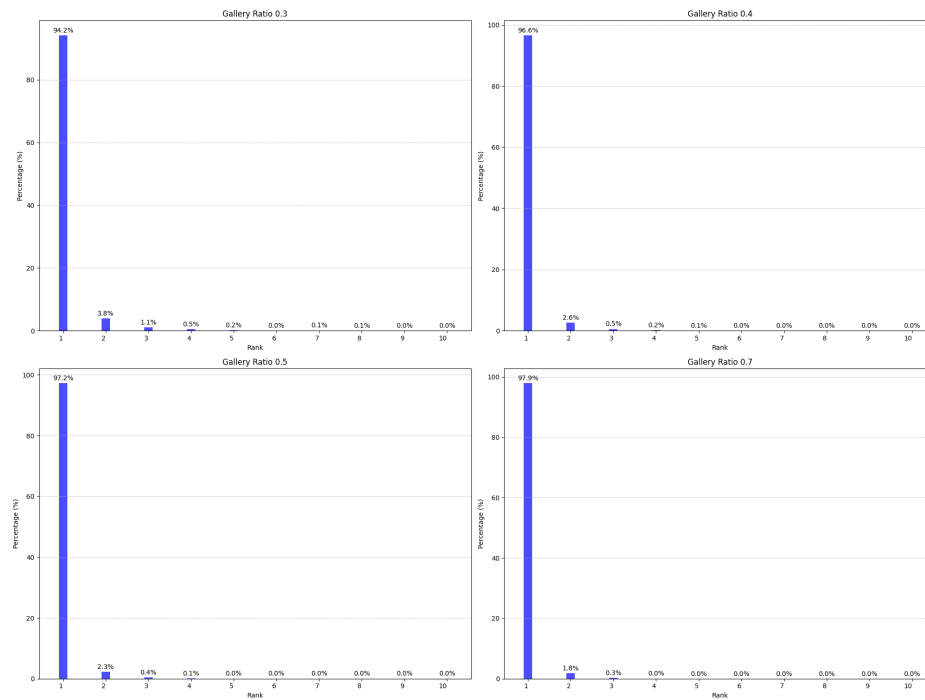
7.0.7 Ranks for FVC2002 Dataset 2 - Peter Kovesi Enhancement



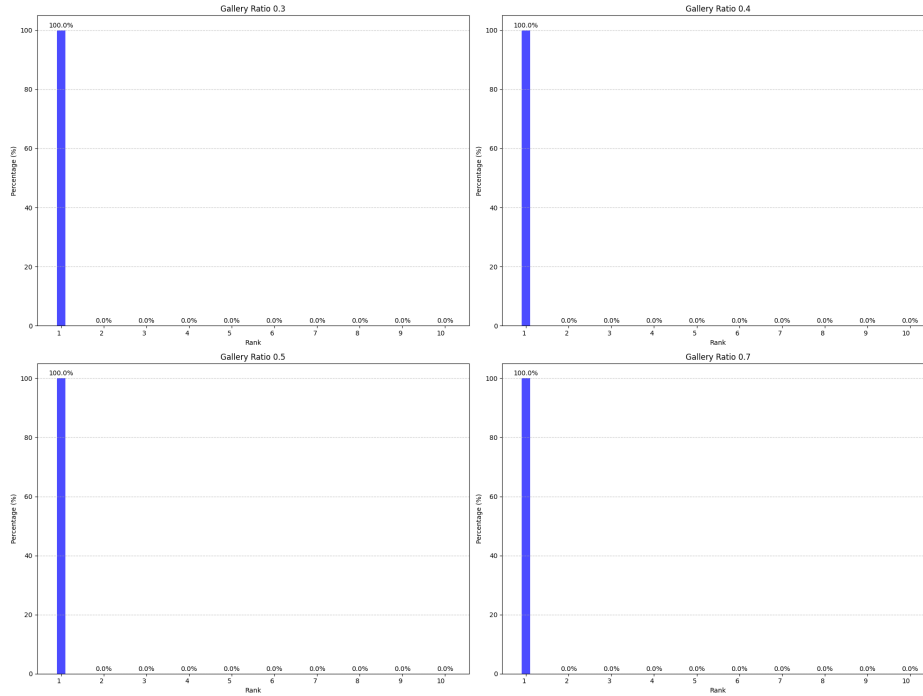
7.0.8 Ranks for FVC2002 Dataset 3 - Peter Kovesi Enhancement



7.0.9 Ranks for FVC2002 Dataset 4 - Peter Kovesi Enhancement



7.0.10 Ranks for Neurotechnology CrossMatch - Peter Kovesi Enhancement



8 Conclusion

The results obtained in this study are encouraging, especially considering that no fine-tuning was applied. This is due to the presence of predefined parameters, as seen in Kovesi's method, and the meticulous fine-tuning process conducted by Cappelli. However, it should be noted that some enhancement steps did not make use of the FVC 2002 dataset (nor NeuroTechnology).

While the performance achieved does not reach levels seen in international competitions, this is largely due to the limited robustness of the matching algorithm. More advanced approaches, such as those that transition from a local matching strategy based on triples to a global matching method [14], or cylinder-based method [3], could yield improved results. Nevertheless, the primary objective of this paper was to demonstrate how specific enhancements, such as those introduced in GMFS compared to Kovesi's method, or the consideration of individual ridge frequencies instead of a global frequency, can impact performance.

A noteworthy aspect is the strong performance of NeuroTechnology. This can be attributed to the nature of the dataset, which was collected under non-adversarial conditions, without excessive distortions or fingerprint cuts. Additionally, while the full dataset contains approximately 500 samples, only 80 were selected for analysis to ensure a reasonable comparison. This choice was also motivated by the computational demands of the matching process, which is not optimized for handling large-scale operations.

Future work will focus on adopting alternative matching strategies, such as those discussed in the referenced document, and further exploring Cappelli's CNN-based enhancement techniques to refine fingerprint processing steps.

References

- [1] Lin Hong, Yifei Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
- [2] P. Kovesi. Image analysis, matlab functions for computer vision, 2003. Accessed: 20 August 2003.
- [3] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32:2128–41, 12 2010.
- [4] Raffaele Cappelli. Unveiling the power of simplicity: Two remarkably effective methods for fingerprint segmentation. *IEEE Access*, 11:144530–144544, 2023.
- [5] Raffaele Cappelli. Exploring the power of simplicity: A new state-of-the-art in fingerprint orientation field estimation. *IEEE Access*, 12:55998–56018, 2024.
- [6] Raffaele Cappelli. No feature left behind: Filling the gap in fingerprint frequency estimation. *IEEE Access*, 12:153605–153617, 2024.
- [7] Raymond Thai. Fingerprint image enhancement and minutiae extraction. Master’s thesis, University of Western Australia, 2003.
- [8] Michael Kass and Andrew P. Witkin. Analyzing oriented patterns. In *International Joint Conference on Artificial Intelligence*, 1985.
- [9] Asker M. Bazen and Sabih H. Gerez. Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:905–919, 2002.
- [10] Michael Garris and R. McCabe. Nist special database 27 fingerprint minutiae from latent and matching tenprint images, 2000-06-01 2000.
- [11] T. Y. Zhang and Ching Yee Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27:236–239, 1984.
- [12] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2nd edition, 2009.
- [13] Farid Benhammadi, Hamid Hentous, Kadda Bey-Beghdad, and Mohamed Aissani. Fingerprint matching using minutiae coordinate systems. In Jorge S. Marques, Nicolás Pérez de la Blanca, and Pedro Pina, editors, *Pattern Recognition and Image Analysis*, pages 529–536, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [14] Miguel Angel Medina-Pérez, Milton García-Borroto, Andres Eduardo Gutierrez-Rodríguez, and Leopoldo Altamirano-Robles. Improving fingerprint verification using minutiae triplets. *Sensors*, 12(3):3418–3437, 2012.