Intelligent Systems with Applications 21 (2024) 200320

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

A novel indexing algorithm for latent palmprints leveraging minutiae and orientation field

Javad Khodadoust [a,*], Raúl Monroy [a], Miguel Angel Medina-Pérez [b], Octavio Loyola-González [c], Worapan Kusakunniran [d], André Boller [e], Philipp Terhörst [f]

[a] School of Engineering and Sciences, Tecnologico de Monterrey, Carr. Lago de Guadalupe Km 3.5, Col. Margarita Maza, Atizapán, Estado de México 52926, Mexico
[b] Kayak Analytics, The Dome Tower, Jumeirah Lake Towers, Dubai, UAE
[c] Stratesys, Calle de Torrelaguna 77, Madrid 28043, Spain
[d] Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom 73170, Thailand
[e] Department of Smart Living and Biometric Technologies, Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
[f] Department of Computer Science, Paderborn University, Paderborn 33098, Germany

ARTICLE INFO

Keywords:
Latent palmprint
Palmprint segmentation
Palmprint indexing
Minutiae importance
Minutiae types
Orientation field

ABSTRACT

Latent palmprints represent crucial forensic evidence in criminal investigations, necessitating their storage in governmental databases. The identification of corresponding palmprints within large-scale databases using an automated palmprint identification system (APIS) is time-consuming and computationally intensive. To address this challenge, this paper introduces an innovative approach: delineating the region of interest (ROI) for palmprint segmentation and presenting a novel indexing algorithm founded on minutiae and the orientation field (OF). Additionally, a novel feature vector is proposed, leveraging minutiae triplets and ellipse properties, marking the pioneering algorithm to consider minutiae importance in palmprint indexing. Significantly, an improved version of an existing palmprint indexing algorithm tailored for latent palmprints is introduced. The study demonstrates the indexing and retrieval of both our feature vectors and those obtained by the improved palmprint indexing algorithm, using two clustering algorithms and locality-sensitive hashing (LSH). The method's robustness is evaluated across three diverse databases with extensive palmprint records. The experimental results underscore the superior performance of our approach compared to current state-of-the-art algorithms.

1. Introduction

In recent years, the significance of biometrics has surged, leading to the establishment of dedicated programs at esteemed universities specializing in this field. Determining the superiority of a specific biometric trait over others is inherently contextual and heavily reliant on the application domain (Maltoni et al., 2021).

A biometric system can operate as a verification system, where an individual claims their identity and the system confirms or rejects it through a one-to-one comparison of captured biometric traits with stored templates. Alternatively, as an identification system, the individual does not need to claim their identity. This system conducts a broader search across the database, comparing captured traits with all stored templates, providing ordered matches based on scores in one-to-many comparisons (Peralta et al., 2015).

Fingerprints and palmprints can be obtained through contact with a sensor, non-contact methods, or basic digital cameras. Latent prints, often found at crime scenes where individuals unintentionally leave their marks, are vital for criminal investigations, serving as crucial evidence to identify perpetrators. Approximately 30% of evidence recovered at crime scenes comprises palmprints according to statistics from law enforcement and forensic agencies (Liu & Qian, 2021, Jain & Feng, 2009, Wang et al., 2011).

Contemporary forensic standards require a palmprint data resolution of 500 pixels per inch (ppi) (Jain & Feng, 2009). Globally, law enforcement and governmental bodies collect biometric information from citizens and visitors. For instance, with a population of nearly

* Corresponding author.
E-mail addresses: khodadoust.j@gmail.com, A01753092@tec.mx (J. Khodadoust), raulm@tec.mx (R. Monroy), m2p@kayak-analytics.com (M.A. Medina-Pérez), octavio.loyola@stratesys-ts.com (O. Loyola-González), worapan.kun@mahidol.edu (W. Kusakunniran), andre.boller@igd.fraunhofer.de (A. Boller), philipp.terhoerst@uni-paderborn.de (P. Terhörst).

https://doi.org/10.1016/j.iswa.2023.200320
Received 11 September 2023; Received in revised form 13 November 2023; Accepted 19 December 2023
Available online 27 December 2023

1,453,000,000 individuals in China (Worldometer), the influx of visitors was substantial before and after COVID-19—145,310,000 in 2019 and 31,980,000 in 2021 (Statista). If a single palmprint image of each hand were acquired from citizens, the database would potentially hold roughly 2,969,960,000 palmprints and associated templates (for both right and left hands).

When a latent palmprint is found at a crime scene, manual identification by forensic experts specialized in latent palmprint analysis becomes impractical, necessitating computer-assisted systems. Even with rapid identification systems like the one proposed by Liu et al. (2013), numerous comparisons against the extensive palmprint database would be required. Utilizing a standard personal computer (PC) similar to Liu et al.'s system, an average comparison time of 115 milliseconds is expected. This extended identification process could take approximately 10.83 years, providing the perpetrator with a substantial window to evade capture.

While fingerprints have well-established and robust classification methods, palmprint classification methods lag significantly (Yang et al., 2011). Despite the potential for faster identification using supercomputers and parallel computing, efforts aim to reduce identification time through alternative techniques, regardless of hardware capabilities. Palmprint indexing is emerging as a crucial strategy to minimize the number of comparisons needed in a database search, presenting a potential solution to this challenge.

Most palmprint indexing algorithms utilize multidimensional vectors that consolidate palmprint features (Khodadoust et al., 2022). The indexing process involves two crucial phases: indexing and retrieval. In indexing, palmprints are systematically mapped to unique identifiers (indices), ensuring samples from the same palm correspond to the same index. During retrieval, the system identifies the most similar palmprints to a given query palmprint, as highlighted in current literature (Schuch, 2019).

In palmprint recognition and indexing, three distinct feature levels are recognized. Level-1 focuses on macro details, including singular points. Level-2 examines minutiae like ridge endings and bifurcations, while level-3 involves micro details, such as sweat pores, often impractical in forensic scenarios due to latent palmprint quality issues (Jain & Feng, 2009). However, the orientation field (OF), a level-2 feature, is commonly used in indexing full palmprints. Crucial features in latent palmprint recognition are typically minutiae and the OF (Yang et al., 2011, Muñoz-Briseño et al., 2015). Visual representations, like those in Fig. 1, display minutiae extracted from a binarized image and the OF for a latent palmprint. Many minutiae extractors use image enhancement for accuracy.

The number of minutiae in a full palmprint far exceeds that found in a rolled fingerprint image, typically by 10 to 14 times, leading to challenges due to an abundance of false minutiae generated by creases (Liu et al., 2013, Zhou et al., 2010). This issue is particularly pronounced in latent palmprints with low quality and complex backgrounds. While methods like Jain and Feng's approach aim to reduce false minutiae in full palmprints, addressing similar issues in latent prints requires precise segmentation algorithms and robust image enhancement techniques (Jain & Feng, 2009, Liu & Qian, 2021). This study introduces a novel and tailored segmentation approach specifically for palmprints. Following segmentation, we enhance palmprints using Jain and Feng's methodology.

Recent research by Makni and Charrier (2020) and Loyola-González et al. (2021) highlighted the importance of individual minutiae in fingerprint verification and identification. However, a common limitation in both studies is their focus on minutiae importance only within query prints, neglecting the impact of specific minutiae in both the query palmprint and its corresponding reference in the database. The accuracy of matching scores could improve by excluding detrimental minutiae from both query and database palmprints. Makni and Charrier gauged minutiae importance based on type but did not examine correlations between type and importance. This approach might inaccurately predict the importance of minutiae, leading to potentially flawed decisions on which minutiae to discard. Loyola-González et al.'s method showcased that removing certain minutiae enhances the matching score, outperforming Makni and Charrier's approach in predicting minutiae importance.

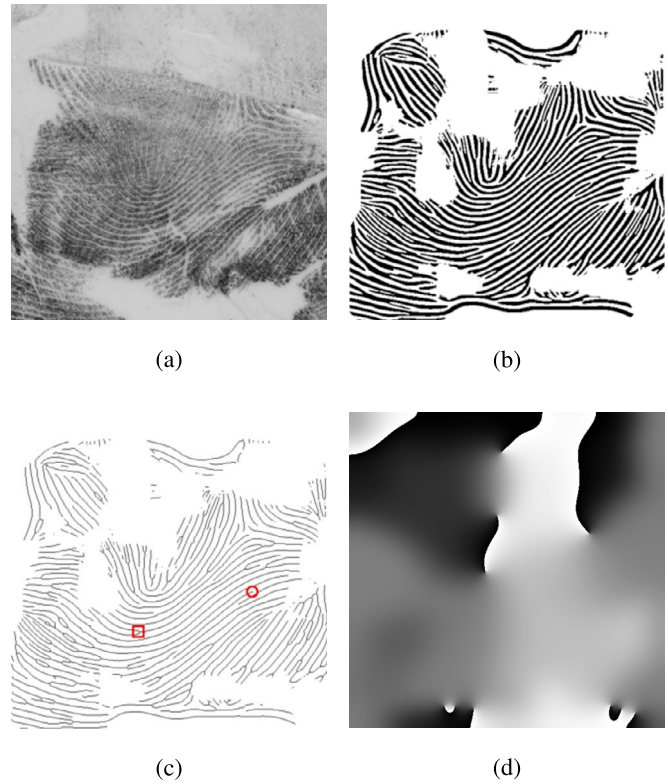


Fig. 1. Palmprint feature extraction: (a) latent grayscale palmprint image, denoted as P000041_l_i_2 in (Morales, Medina-Pérez et al., 2014), (b) binarized image derived from (a) using VeriFinger, (c) ridge skeleton image generated from (a) using VeriFinger, where the circle and square represent a ridge ending minutia and a ridge bifurcation minutia, respectively, and (d) the OF obtained from Khan et al.'s approach (Khan et al., 2016).

In this study, we analyze minutiae importance in both query and corresponding palmprints, classifying them into six categories, delineating their importance and types. Unfortunately, the absence of a widely accepted quality assessment for latent palmprints impedes exploration into the potential relationship between minutiae importance and their quality (Khodadoust et al., 2022). Additionally, we introduce a precise indexing approach tailored for latent palmprints, utilizing minutiae to form feature vectors. Integration of the OF effectively reduces the error rate (ER). The key contributions of this study can be summarized as follows:

i. A novel generative adversarial network (GAN)-based approach for palmprint segmentation, aimed at achieving a region of interest (ROI), is presented in this research. The methodology involves initially transforming the palmprint image into the frequency domain, followed by several operations that result in a binary image. This binary image is used as input data for the GAN model, which generates a binary mask to effectively segment the palmprint image.
ii. A novel minutiae-based indexing method is introduced for latent palmprints. Leveraging the intricacies of ellipses and the importance of minutiae, this approach notably reduces the ER. Its superior efficacy, compared to alternative strategies, is further enhanced through the integration of the OF, serving to reduce the ER and refine the candidate list. Notably, this method marks the pi-

oneering utilization of the OF in the domain of latent palmprint indexing.

iii. By meticulously assessing the influence of each minutia on the matching score, we categorize them into six distinct classes: neutral ridge ending, neutral ridge bifurcation, removable ridge ending, removable ridge bifurcation, irremovable ridge ending, and irremovable ridge bifurcation. This classification framework offers valuable insights into the intricate relationship between minutiae importance and their specific types.

iv. Our methodology uniquely leverages minutiae importance to assign varying weights to our proposed feature vectors, an approach that effectively curtails the ER. Remarkably, there exists no prior work that has undertaken the incorporation of minutiae importance in the context of latent palmprint indexing.

The subsequent sections of this paper are structured as follows: Section 2 presents a comprehensive review of related works in the field. Section 3 delineates the specifics of our methodology. In Section 4, the experimental setup is elaborated upon, accompanied by an in-depth analysis of the experimental findings. Lastly, Section 5 encapsulates the conclusion of this paper.

2. Related work

In this section, we initially review the existing methods for predicting minutiae importance. Subsequently, we delve into a review of the prevailing indexing algorithms utilized for palmprint indexing.

2.1. The prediction of minutiae importance

The impact of minutiae on the accuracy of fingerprint and palmprint recognition varies. The exclusion of certain minutiae has minimal influence on the matching score, while removing others can result in either a decrease or an increase in the overall matching score.

In their work, Makni and Charrier (2020) introduced a confidence metric known as the minutia confidence index (MiCI), which enables the prediction of the impact and significance of individual minutiae on the accuracy of AFISs. Their approach involves utilizing minutiae attributes such as coordinates, orientation, and type to assess the relative importance of each minutia compared to others. By employing the MiCI, they were able to identify and subsequently eliminate minutiae with the lowest predicted importance.

Loyola-González et al. (2021) conducted an investigation into the potential positive or negative influence of individual minutiae on the precision of AFISs, particularly concerning latent fingerprints. Their research highlighted the significant impact that removing certain minutiae could have on the accuracy of fingerprint identification. To further explore this aspect, they compiled a dataset and applied various machine learning models for training purposes. The central objective of their study was to develop a predictive framework capable of assessing the importance of individual minutiae in determining the accuracy of an AFIS.

2.2. Palmprint indexing methods

This section provides an overview of the currently available techniques for indexing palmprints.

Li et al. (2005) introduced a texture-based approach aimed at quantifying both local and global attributes of human palmprints using numerical values. Depending on the characteristics of individual palmprints, global features alone may suffice for differentiation, while others require the incorporation of local features for effective recognition. In their retrieval process, they initially utilized global attributes to retrieve potential candidates. Subsequently, local features were employed on these candidates to narrow down the list.

Paliwal et al. (2010) introduced an innovative indexing approach centered around the generation of matching score vectors, which served as indices. These matching score vectors encapsulate multiple matching scores, potentially resulting in high-dimensional representations. To manage this complexity, the vector approximation (VA+) file was leveraged for indexing purposes. In the retrieval process, the authors employed dynamic index structures to efficiently reduce the scope of vectors that necessitate consideration. Furthermore, the integration of palmprint texture was incorporated to enhance the overall palmprint recognition performance.

Yang et al. (2011) introduced an indexing technique tailored for impression palmprints. This approach involved the utilization of both palmprint alignment and coarse matching strategies. Initially, the authors aligned palmprints onto a standardized coordinate system by employing the OF. Following this alignment, a comparative analysis was conducted between the OF and the density map. This process led to the derivation of a candidate list, refining the pool of potential matches.

Muñoz-Briseño et al. (2015) introduced an indexing approach centered around minutiae triplets. The authors put forth the concept of expanded triangles, designed to accommodate the inherent challenges posed by palmprint rotation and geometric distortions. This method aimed to enhance the robustness of indexing in palmprint recognition.

Kavati et al. (2017) introduced an indexing technique centered on the utilization of matching scores. Their approach hinged on the observation that the matching scores of distinct impressions of a palm against another palmprint image, denoted as s, exhibit remarkable proximity. This proximity served as the basis for constructing an index table where these matching scores were employed as keys. During the retrieval phase, when a query palmprint is matched with s, the corresponding matching score is sought in the index table. Through a voting mechanism, palmprints with the most similar scores are identified as candidate matches. A subsequent step involves the filtering of false candidates to enhance the accuracy of the retrieval process.

Arora et al. (2021) introduced an innovative palmprint indexing technique named PalmHashNet, which is rooted in neural network principles. To effectively guide the training process, they incorporated an additive margin loss mechanism, enabling the extraction of metric-based features. The key premise underlying indexing methods is the generation of features that demonstrate substantial intra-class variability while minimizing inter-class variation. In pursuit of this goal, they integrated the softmax loss with an additional margin in the loss function. Furthermore, they harnessed the power of locality-sensitive hashing (LSH) and k-means clustering algorithms, strategically applied within both the indexing and retrieval phases of the process.

Khodadoust et al. (2022) introduced an indexing approach leveraging the concept of the middle of the triangle's side (MTS). Their methodology involved the utilization of Delaunay triangulation of order k applied to minutiae, which in turn facilitated the derivation of MTSs. These MTSs were harnessed as fundamental components for indexing and retrieving palmprints. Notably, their method encompassed the generation of a quality map that factored in the quality assessment of minutiae, effectively integrating it within their indexing framework.

With the exception of the approach presented by Khodadoust et al. (2022), the entirety of existing palmprint indexing techniques have been primarily tailored for full palmprints. Addressing the extraction of latent prints, which often emerge as partial, low quality representations and entail intricate backgrounds, poses a formidable challenge. Moreover, it is worth noting that Khodadoust et al. did not incorporate minutiae importance, a factor that significantly influences palmprint recognition. Thus, the subsequent section will introduce the framework of our pioneering indexing algorithm, specifically tailored for latent palmprints.

3. Methodology

To design and implement an indexing method, two phases must be taken into consideration. In the first phase, referred to as indexing, the method extracts biometric features, such as minutiae and the OF in palmprints. Subsequently, these features are utilized to create feature vectors, which are then stored within an index table. In the second phase, known as the retrieval phase, similar to the indexing phase, features of a query biometric trait are extracted, and feature vectors are generated. Ultimately, a search strategy is employed to identify the most similar feature vectors stored in the database in relation to the feature vectors of the query biometric trait (Cappelli et al., 2011).

In this study, we have introduced a minutiae-based indexing method tailored for latent palmprints. Leveraging the attributes of ellipses and minutiae importance, we have achieved a noteworthy reduction in the ER. Furthermore, by assigning distinct weights to feature vectors based on minutiae importance, we have established a scoring mechanism that ranges from 0 to 1 for individual votes. Additionally, we have improved the indexing approach originally presented by Kavati et al. (2017), enabling its applicability to partial and latent palmprints.

3.1. Indexing phase

When selecting palmprint features, it becomes imperative to focus on those attributes that possess the capability to effectively filter out a substantial portion of false candidates. Additionally, the dimensionality of feature vectors plays a crucial role in enabling swift comparisons, thereby rendering the methodology viable for extensive databases. In this study, we opted to incorporate both minutiae and OF as our chosen palmprint features.

Mehmood et al. (2023) introduced a palmprint segmentation approach incorporating a circular Gaussian bandpass filter, inspired by the technique proposed by by Sutthiwichaiporn and Areekul (2013), Otsu's method, and a Gaussian filter with a standard deviation (σ) of 20 for extensive blurring. However, our investigation revealed limitations in accurately delineating the border between foreground and background using this method. As an alternative to employing Otsu's method for image binarization and subsequent smoothing, we devised a generative adversarial network (GAN) to enhance the accuracy of ROI extraction. In our GAN framework, the initial step involves resizing the binary image to dimensions of 256×256, followed by utilizing an encoder and a decoder to generate the desired ROI. Training the network involved using a combination of loss functions, including binary cross-entropy (BCE) and soft dice losses (Bertels et al., 2019).

The BCE serves as a loss function applied in binary classification duties, originating from information theory. It gauges the variance between two probability distributions. In the realm of binary classification, the BCE assesses the dissimilarity between predicted probabilities and real binary labels (0 or 1). The BCE equation is as follows:

$$BCE(g,p) = -\frac{1}{N}\sum_{i=1}^{N}(g_i.\log(p_i) + (1-g_i).\log(1-p_i)), \tag{1}$$

where g represents the ground truth (true label, 0 or 1), p denotes the predicted probability, N signifies the number of samples or instances, and g_i and p_i correspond to the individual target and predicted probability for the i-th sample. This equation calculates the loss for each individual sample and then averages the losses across all samples (the $\frac{1}{N}\sum$ part).

The BCE formula penalizes the model more when the predicted probability (p) significantly deviates from the actual target (g). As the prediction (p) approaches the true label (g), the loss tends toward zero.

To calculate the Soft Dice, the Dice coefficient is initially obtained. This metric is commonly used in training neural networks due to its compatibility with gradient-based optimization methods. Soft Dice is particularly beneficial when dealing with continuous or fuzzy regions, as it offers a smooth gradient that the BCE might lack.

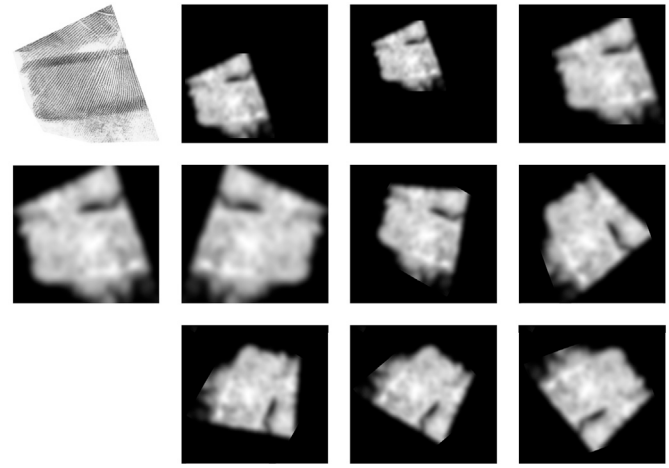


Fig. 2. Data augmentation: the top image in the left column is P000007_l_h_1 from (Morales, Medina-Pérez et al., 2014), while the bottom image in the left column represents the blurred binary image. The remaining images showcase the outcomes of the augmentation process.

The Soft Dice is obtained as follows.

$$Dice = \frac{2\sum_{i=1}^{N} g_i p_i + \epsilon}{\sum_{i=1}^{N} g_i + \sum_{i=1}^{N} p_i + \epsilon}, \tag{2}$$

$$Dice_b = \frac{2\sum_{i=1}^{N}(1-g_i)(1-p_i) + \epsilon}{\sum_{i=1}^{N}(1-g_i) + \sum_{i=1}^{N}(1-p_i) + \epsilon}, \tag{3}$$

$$L_{SoftDice} = 1 - \frac{Dice + Dice_b}{2}, \tag{4}$$

Annotations are represented by variables $g_i \in \{0,1\}$ and $p_i \in \{0,1\}$, and probability maps are utilized to prevent division by zero. Moreover, the variable $\epsilon \in \mathbb{R}$ represents a small constant, typically added for numerical stability to avoid division by zero.

In this study, the loss of our model is determined by Eq. (5).

$$Loss = W_b BCE + W_s L_{SoftDice}. \tag{5}$$

where W_b and W_s are weights between 0 and 1, and $W_b + W_s = 1$, representing the weights for BCE and $L_{SoftDice}$, respectively.

Dice is typically utilized to assess the foreground, while BCE is applied to address the binary segmentation problem concerning the foreground-background relationship. Soft Dice, serving as a differentiable loss function, is specifically employed in training optimization, especially effective in scenarios with fuzzy boundaries or continuous segmentation tasks. Both BCE and Soft Dice are applicable for training segmentation models that cater to both foreground and background regions. This approach aims to guide the network in precisely categorizing each pixel as part of the palmprint or pertaining to the background.

It is worth noting that we have employed augmentation techniques, including tasks such as rotation, horizontal flipping, and image resizing. These methods are employed to augment the diversity of the image dataset, effectively expanding the range of training samples. Fig. 2 provides an example of augmentation.

The architecture of our GAN is depicted in Fig. 3. Following the acquisition of ROI, an additional step involves the exclusion of foreground pixels with a standard deviation lower than a predetermined threshold value t, which we have set at 0.1 based on the research conducted by Kovesi (2022). Employing a thorough analysis of standard deviation, we accurately determined and assigned a value of 0 to pixels located within the border region, characterized by a lower standard deviation. Improving the quality of ROI leads to a decrease in the count of erroneous minutiae. Fig. 4 illustrates the results of each step in our segmentation method, while Fig. 5 compares our method with the approach proposed by Mehmood et al. (2023).

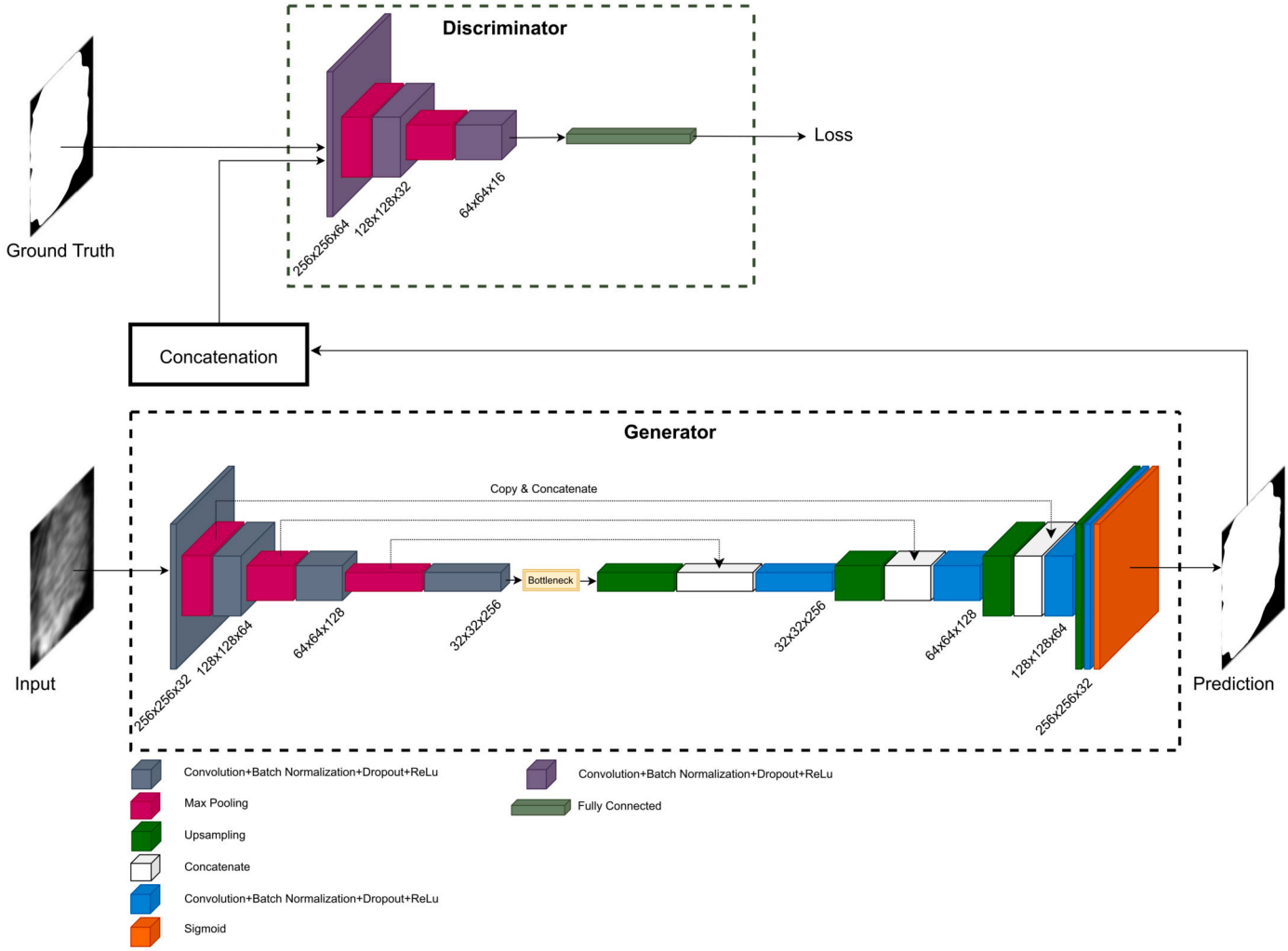


Fig. 3. Overall architecture of our proposed method.

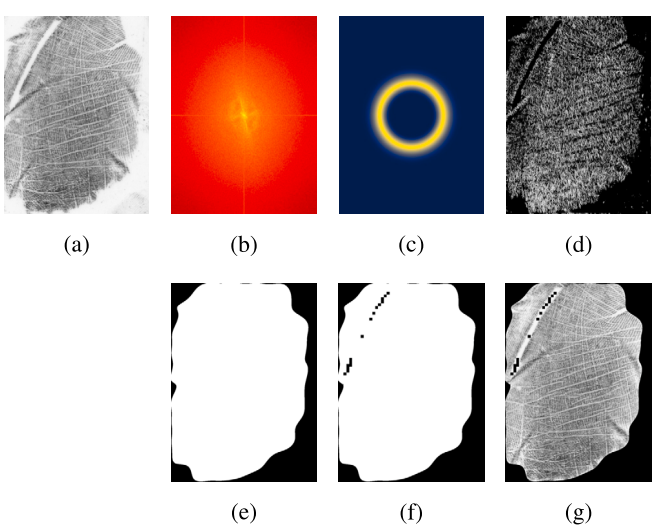


Fig. 4. Our palmprint segmentation method: (a) original latent grayscale palmprint image, identified as P000026_l_t_1 in (Morales, Medina-Pérez et al., 2014), (b) frequency image derived from (a), (c) circular Gaussian bandpass filter introduced by Sutthiwichaiporn and Areekul (2013), (d) resulting image obtained by multiplying (b) and (c), transformed to the spatial domain, (e) ROI acquired through our GAN, (f) ROI after excluding foreground pixels with a standard deviation below 0.1, and (g) the segmented image.

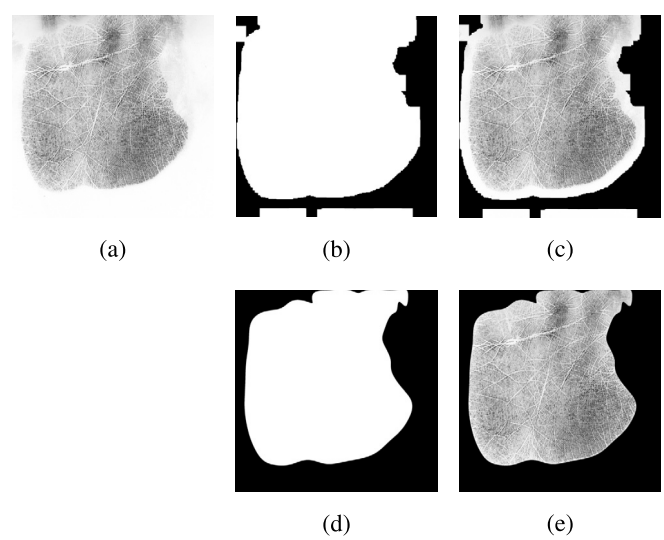
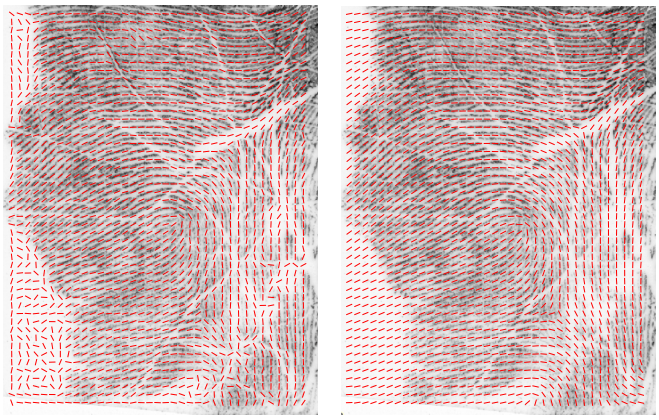


Fig. 5. Comparison of our palmprint segmentation method and the approach proposed by Mehmood et al. (2023): (a) original palmprint (4_l_2 in (THU-PALMLAB)), (b) binary mask acquired through Mehmood et al.'s method, (c) the segmented image by Mehmood et al.'s method, (d) binary mask acquired through our GAN, and (e) the segmented image obtained by our method.



(a) (b)

Fig. 6. Comparison of the OF estimated using two distinct methods on a latent palmprint (P000033_l_h_2 in (Morales, Medina-Pérez et al., 2014)): (a) OF estimated by Kovesi (2022), and (b) the OF estimated by Khan et al. (2016). Notably, Khan et al.'s method demonstrates better performance in estimating the OF within noisy and creased regions compared to Kovesi's method.

Following the segmentation process applied to the input image, yielding ROI, OF estimation was pursued. Various techniques for OF estimation exist in the literature (Maltoni et al., 2021). Among these, we opted to utilize the approach introduced by Khan et al. (2016). This choice is rooted in its capability to accurately estimate the OF, particularly within regions marked by creases (Khodadoust et al., 2022). Fig. 6 presents a comparison between two distinct methods for estimating the OF on a latent palmprint. Notably, the method proposed by Khan et al. (2016) exhibited superior performance in estimating the OF in noisy and creased regions, in contrast to the Kovesi's method (Kovesi, 2022).

Subsequent to the OF acquisition, we implemented the technique proposed by Soleimani and Ahmadi (2018) to align impression palmprints and harmonize them within a common coordinate system. This selection stems from the fact that the Soleimani and Ahmadi's method utilizes the pre-estimated OF to align palmprint images, thereby mitigating the computational load. To our knowledge, the method introduced by Soleimani and Ahmadi represents the initial effort to register impression palmprints. In contrast, alternative approaches, like the Balwant et al.'s method (Balwant et al., 2015), were intended for contactless palmprints and required all fingers of the hand for aligning palmprint images. As depicted in Fig. 7, the outcomes of the aforementioned method on an impression palmprint exhibit the absence of fingertip details and inaccuracies in detecting finger valleys. Conversely, the Soleimani and Ahmadi's method aptly aligns the palmprint image with precision.

Following that, we employed the enhancement method proposed by Jain and Feng (2009) to improve the palmprint image. To extract minutiae, we adopted the approach presented by Zhao and Tang (2007), which includes a strategy to reduce the occurrence of erroneous minutiae in the foreground.

Consequently, employing our segmentation method prevents the inclusion of erroneous minutiae in the background, while utilizing the minutiae extraction technique introduced by Zhao and Tang (2007) serves to diminish the count of inaccurate minutiae present in the foreground.

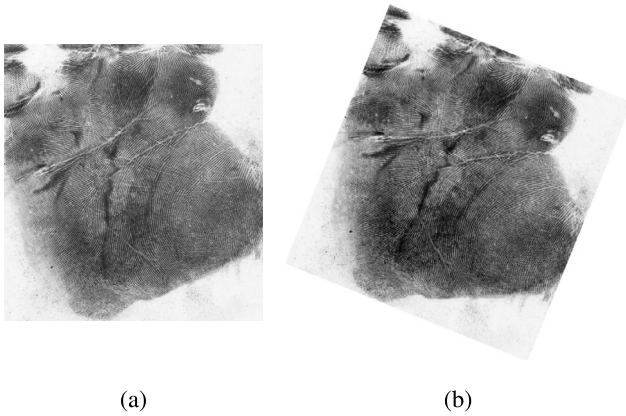
Upon estimating the OF and extracting minutiae, we chose to exclusively focus on the utilization of minutiae for constructing index vectors. This decision was influenced by the enhanced indexing and retrieval capabilities attributed to minutiae, as opposed to the OF, as emphasized by Yang et al. (2011). To gauge the significance of these minutiae, we adapted Loyola-González et al.'s methodology (Loyola-González et al., 2021), customizing it to align with our objectives. While the original approach classified minutiae into "higher" and "lower" categories to gauge importance, we extended this binary framework to accommodate multi-class scenarios. In our context, minutiae were classified into six distinct classes, encompassing neutral ridge endings, neutral ridge bifurcations, removable ridge endings, removable ridge bifurcations, irremovable ridge endings, and irremovable ridge bifurcations. This expanded classification scheme serves to elucidate the nuanced relationship between minutiae types and their respective importance levels. Loyola-González et al.'s approach is grounded in the assessment of a specific minutia. The method computes the count of neighboring minutiae within six distinct radii, namely, 15 pixels, 30 pixels, 45 pixels, 60 pixels, 75 pixels, and 90 pixels, and additionally determines the distances to the six closest neighbors. Consequently, a set of 12 distinctive features is derived. To enhance this feature set, we considered the minutia confidence index (MiCI), as proposed by Makni and Charrier (2020). The MiCI quantifies minutiae with a confidence value, using a reference minutia m_{ref} as a basis, leading to the assignment of a MiCI value to each minutia. This comprehensive approach enabled us to predict minutiae importance across latent query palmprints, palmprints in the sample image set, and full palmprints stored in the database.

To establish minutiae importance ground truth within our training set, we followed a specific approach. In our indexing procedure, we considered n minutiae, labeled as $m_1, m_2, \ldots, m_n$, in the latent query palmprint q, and n' minutiae, labeled as $m'_1, m'_2, \ldots, m'_{n'}$, in the full palmprint f. We undertook a two-step strategy to address potential disparities in matching scores. Initially, we computed the matching score between q and f, referred to as $match(q, f)$, and subsequently, the matching score between f and q, denoted as $match(f, q)$. To determine minutiae importance, we employed a process where we obtained the reference matching score $match(q, f)$ and iteratively removed each minutia from q, recalculating the matching score for the remaining minutiae. The difference between this recalculated score and the reference score indicated whether a minutia was removable (positive difference), irremovable (negative difference), or neutral (difference equals 0). This analysis was repeated for all minutiae in q. Notably, for subsequent minutiae, we reintroduced previously removed minutiae before calculating the matching score, leading to a total of n matching tasks.

In the subsequent phase, we proceeded with the methodology as follows: Initially, we computed the reference matching score $match(f, q)$ and then successively removed m'_1 from f. Subsequent to this removal, we obtained $match(f, q)$ for the remaining minutiae within f. The difference between this obtained score and the reference matching score was calculated. Mirroring the procedure of the initial step, if the result proved positive, the minutia was labeled as removable. Conversely, if the result was negative, the minutia was designated as irremovable. If the result equated to 0, the minutia was categorized as neutral. This process was systematically repeated for all other minutiae encompassed within f.

To derive the ground truth for minutiae importance within the training set, in our improved version of Kavati et al.'s method (Kavati et al., 2017), we executed the described procedure involving the query palmprints and the palmprints stored within the sample image set. Additionally, we applied this procedure between the palmprints archived in the sample image set and the full palmprints stored within the database.

The determination of minutiae importance through the adapted Loyola-González et al.'s method (Loyola-González et al., 2021) is contingent upon the specific minutiae matcher employed. Notably, minutiae importance varies when utilizing distinct matchers. For instance, the importance calculated using the Bozorth3 matcher (NBIS) differs from that computed using the minutia cylinder-code (MCC) matcher (Cappelli et al., 2010). It is important to emphasize that a consistent minutiae matcher must be employed, precluding the use of different matchers concurrently. By incorporating the minutiae importance factor, we achieved a reduction in the ER and a subsequent decrease in the number of candidates within the candidate list. Our methodology hinged on the use of the Bozorth3 matcher for palmprint matching. Notably, certain alternative minutiae matchers, such as VeriFinger, are incom-



(a) (b)

Fig. 7. Palmprint alignment: (a) a palmprint image (P000040_l in (Morales, Medina-Pérez et al., 2014)) and (b) the aligned image obtained by the Soleimani and Ahmadi's method (Soleimani & Ahmadi, 2018).

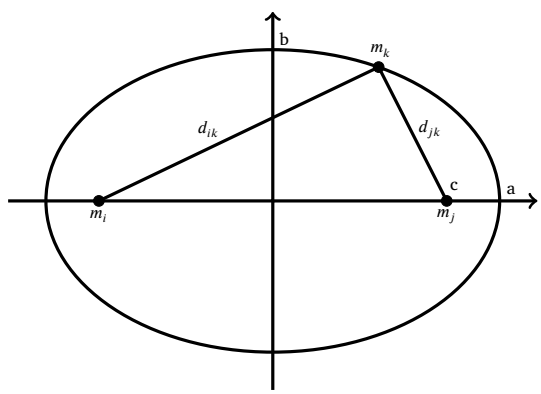


Fig. 8. Illustration of an ellipse and its particulars. The figure depicts an ellipse accompanied by m_i, m_j, and m_k, representing three distinct minutiae. d_{ik} and d_{jk} signify the distances between m_i and m_k, and between m_j and m_k respectively. The parameter c designates the focal distance of the ellipse, while a and b respectively denote the major and minor axes of the ellipse.

patible with our approach. This arises from the inability to remove a minutia from a palmprint and then compute a matching score, as the output files, namely 'query.template' and 'corresponding.template', are encrypted. Another well-known matcher, MCC, should also be avoided within our method, given that removing a minutia adversely affects all cylinders associated with it.

In our indexing approach, we focused on minutiae triplets derived through Delaunay triangulation of order k. It has been demonstrated that this particular triangulation method outperforms alternatives (Khodadoust et al., 2022). Notably, a significant correlation exists between triangles and ellipses (Khodadoust & Khodadoust, 2017a). To establish indices, we opted for ellipses and their inherent properties instead of triangles. This choice is substantiated by research indicating the superiority of ellipses over triangles for the indexing and retrieval of fingerprints (Khodadoust & Khodadoust, 2017b). The ellipse, as evidenced by Fig. 8 and Eqs. (6) and (7), encompasses various attributes that render it an optimal choice for our approach.

$$a^2 = b^2 + c^2, \tag{6}$$

$$a = \frac{d_{ik} + d_{jk}}{2}. \tag{7}$$

Within each triangle, we correspondingly define three ellipses. This study employs the area and perimeter of these ellipses as key metrics. The calculation for the area of an ellipse is presented in Eq. (8).

$$A = \pi ab. \tag{8}$$

Until now, no equation exists for directly calculating the perimeter of an ellipse. Nevertheless, there are several approximations that can be utilized for this purpose. In this study, we employed Ramanujan's approximation (Villarino, 2005) to determine the perimeter of an ellipse (Eqs. (9) and (10)).

$$\eta = \frac{(a-b)^2}{(a+b)^2}, \tag{9}$$

$$p \approx \pi(a+b)(1 + \frac{3\eta}{10 + \sqrt{4 - 3\eta}}). \tag{10}$$

It is worth noting that the area and perimeter of triangles lack distinctiveness, often resulting in a considerable number of false candidates being included. However, our experiments have demonstrated that the area and perimeter of ellipses possess a level of distinctiveness that allows for the listing of candidates, a significant proportion of which are accurate. Our feature vectors were formulated using Eq. (11).

$$\begin{aligned} V_1 &= (A_1, P_1, \theta_{l1}, \theta_{r1}, \theta_{lr1}, S), \\ V_2 &= (A_2, P_2, \theta_{l2}, \theta_{r2}, \theta_{lr2}, S), \\ V_3 &= (A_3, P_3, \theta_{l3}, \theta_{r3}, \theta_{lr3}, S). \end{aligned} \tag{11}$$

where A_1, A_2, and A_3 represent the areas of ellipses, A_1 pertains to the ellipse with focal points positioned on the longest side of the triangle, A_2 pertains to the ellipse with focal points situated on the middle side of the triangle, and A_3 corresponds to the ellipse with focal points placed on the shortest side of the triangle. θ_l represents the interior and left angle formed by the side of the triangle where the foci are positioned and the side where the foci are not situated. Similarly, θ_r denotes the interior and right angle between the side where the foci are placed and the side where they are not. θ_{lr} indicates the angular difference between the left and right minutiae on the side of the triangle where the foci are located. S denotes the triangle's sign value, calculated using Eq. (12). Note that x_i and y_i represent the coordinates of minutia m_i located at one vertex of the triangle, while x_j and y_j denote the coordinates of minutia m_j situated at another vertex of the triangle. Similarly, x_k and y_k correspond to the coordinates of minutia m_k placed at the remaining vertex of the triangle. The pseudo-code for constructing our feature vectors is outlined in Algorithm 1.

$$S = \begin{cases} 1 & if \quad x_i(y_j - y_k) + x_j(y_k - y_i) + x_k(y_i - y_j) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

After generating the feature vectors, an essential preprocessing step known as normalization becomes necessary. This is due to the fact that the values within a feature vector typically span different ranges. For instance, in our feature vector, the values and ranges for area, perimeter, angle, and triangle sign differ. Consequently, it becomes imperative to normalize and rescale the values across all feature vectors to a predetermined range. Various normalization techniques exist in the literature, and for our approach, we selected min-max normalization. This method involves transforming the values of all feature vectors to fit within the range $[\alpha, \beta]$ (Goldstein & Uchida, 2016) (refer to Eq. (13)). Opting for min-max normalization is primarily attributed to the fact that we do not require knowledge about the distribution of feature vectors, and furthermore, the original distribution shape of the feature vectors remains preserved.

$$x_{norm} = \frac{(x - min(x))(\beta - \alpha)}{max(x) - min(x)} + \alpha. \tag{13}$$

where the original value of the variable intended for normalization is represented by x, $min(x)$ designates the minimum value of x within our dataset, and $max(x)$ signifies the maximum value of x within our dataset. Defining values for the desired range of the normalized variable are provided by parameters α and β, and the normalized value of the variable x is denoted as x_{norm}. The values of α and β determine

the lower and upper bounds of the desired range for the normalized variable. The equation, responsible for scaling and shifting the original values of x to align within this specified range, is then employed. Empowerment to regulate the range of the normalized values according to specific requirements is achieved through the selection of α and β.

It is important to note that we employed an enhanced variant of the k-means clustering algorithm, namely the k-harmonic means and overlapping k-means algorithms (KHM-OKM) (Khanmohammadi et al., 2017), to store feature vectors within the database. This approach exhibits insensitivity to outliers and permits a feature vector to belong to two or more clusters, thus enabling the retrieval of a significant number of accurate candidates. The initial step involves determining the cluster centers using Eqs. (14) and (15).

$$z_j = \frac{\sum_{i=1}^{n} A_{ij} v_i}{\sum_{i=1}^{n} A_{ij}}, \tag{14}$$

$$A_{ij} = m(z_j|v_i) w(v_i), \tag{15}$$

where v_i denotes the i-th feature vector, $w(v_i)$ represents the weight of v_i, and $m(z_j|v_i)$ signifies the membership of v_i to the centroid of cluster j.

$$w(v_i) = \frac{\sum_{j=1}^{k} ||v_i - z_j||^{-p-2}}{(\sum_{j=1}^{k} ||v_i - z_j||^{-p})^2}, \tag{16}$$

$$m(z_j|v_i) = \frac{||v_i - z_j||^{-p-2}}{\sum_{j=1}^{k} ||v_i - z_j||^{-p-2}}, \tag{17}$$

where p is treated as a free parameter ($p \geq 2$), and k represents the number of clusters. The parameter p plays a crucial role in determining the weight $w(v_i)$ and the membership $m(z_j|v_i)$ of each feature vector v_i to the clusters. It controls the influence of the distances between feature vectors and cluster centroids in the calculation of both weights and memberships.

The selection of p allows for flexibility in adapting the algorithm to different scenarios. A higher value of p emphasizes larger distances, potentially leading to more localized clusters, while a lower value considers a broader range of distances, accommodating clusters with varying spatial distributions.

Algorithm 1 Pseudo-code for generating feature vectors.

Input: Minutiae set M
Output: Feature vector V
$V \leftarrow \emptyset$;
$DTk \leftarrow Delaunay\_Triangulation\_of\_order\ k\ (M)$;
for each $\triangle A_i B_j C_k \in DTk$ **do**
  $E_{ij} \leftarrow Ellipse\ (A_i B_j)$; ▷ Focal points of the ellipse E_{ij} positioned along the side $A_i B_j$
  $E_{jk} \leftarrow Ellipse\ (B_j C_k)$;
  $E_{ik} \leftarrow Ellipse\ (A_i C_k)$;
  $Area_{ij} \leftarrow Area\ (E_{ij})$; ▷ The area of the ellipse (E_{ij})
  $Area_{jk} \leftarrow Area\ (E_{jk})$;
  $Area_{ik} \leftarrow Area\ (E_{ik})$;
  $Perimeter_{ij} \leftarrow Perimeter\ (E_{ij})$; ▷ The perimeter of the ellipse (E_{ij})
  $Perimeter_{jk} \leftarrow Perimeter\ (E_{jk})$;
  $Perimeter_{ik} \leftarrow Perimeter\ (E_{ik})$;
  $S \leftarrow Sign\ (\triangle A_i B_j C_k)$; ▷ The sign of $\triangle A_i B_j C_k$
  $\theta_{lij} \leftarrow \theta_i$;
  $\theta_{rij} \leftarrow \theta_j$;
  $\theta_{ljk} \leftarrow \theta_j$;
  $\theta_{rjk} \leftarrow \theta_k$;
  $\theta_{lik} \leftarrow \theta_k$;
  $\theta_{rik} \leftarrow \theta_i$;
  $\theta_{lrij} \leftarrow |\theta_{m_i} - \theta_{m_j}|$;
  $\theta_{lrjk} \leftarrow |\theta_{m_j} - \theta_{m_k}|$;
  $\theta_{lrik} \leftarrow |\theta_{m_i} - \theta_{m_k}|$;
  $V_{ij} \leftarrow (A_{ij}, P_{ij}, \theta_{lij}, \theta_{rij}, \theta_{lrij}, S)$;
  $V_{jk} \leftarrow (A_{jk}, P_{jk}, \theta_{ljk}, \theta_{rjk}, \theta_{lrjk}, S)$;
  $V_{ik} \leftarrow (A_{ik}, P_{ik}, \theta_{lik}, \theta_{rik}, \theta_{lrik}, S)$;
  $V \leftarrow V \cup V_{ij} \cup V_{jk} \cup V_{ik}$;
return V;

The choice of p can be guided by the characteristics of the dataset and the desired behavior of the clustering algorithm.

Subsequently, the OKM method was initialized with these outcomes using Eq. (18).

$$z_j = \frac{1}{\sum_{v_i \in \pi_j} \frac{1}{\delta_i^2}} \sum_{v_i \in \pi_j} \frac{1}{\delta_i^2} . (\delta_i \times v_i - \sum_{z_j \in \pi(v_i)/z_j} z_j), \tag{18}$$

where $\pi(v_i)$ represents a list of clusters to which v_i belongs. Here, $\delta_i = |\pi(v_i)|$ indicates the total count of clusters to which v_i is associated.

In the subsequent step, the ultimate cluster centers were derived through the OKM method. Ultimately, clusters to which each feature vector belongs were determined using Eq. (19).

$$m(z_j|v_i) = \frac{||v_i - z_j||^2}{\sum_{j=1}^{k} ||v_i - z_j||^2}. \tag{19}$$

In addition, we have improved and adapted the technique introduced by Kavati et al. (2017). In the indexing phase, the original method proposed by Kavati et al., which involves normalizing matching scores within the range [0, 100] for palmprints in the sample image set and those in the database, is not applicable to partial and latent palmprints. Consequently, it necessitates modification. To compute matching scores, we incorporated minutiae importance and excluded certain removable minutiae—specifically, those minutiae that detrimentally impact matching scores and possess the lowest MiCI.

Furthermore, we applied the approach proposed by Krish et al. (2014) to both the palmprints in the sample image set and those in the database. The rationale for adopting Krish et al.'s method stemmed from its capability to leverage the OF exclusively, enabling it not only to align partial palmprints within full palmprints but also to align smaller partial palmprints within larger ones. Our assessment of the OF similarity between a palmprint in the sample image set and the palmprints in the database led us to set the corresponding dimension of the feature vector to 0 if the similarity between each palmprint pair fell below a certain threshold. For palmprints with a similarity exceeding or equal to the threshold, the Krish et al.'s method performed palmprint alignment by embedding the palmprint from the sample image set into the full palmprint within the database, discarding minutiae that lay outside this region. Subsequently, new matching scores were obtained. Following this, we normalized the matching scores and rescaled them to the range (0, 1) using the softmax function. To achieve binarized matching scores, we employed linear discriminant analysis (LDA) in conjunction with predefined thresholds. Loyola-González et al. (2021) demonstrated that LDA yielded superior outcomes compared to most binary classifiers. Given that our feature vectors were characterized by fixed-length, binary, and sparse properties, we harnessed LSH (Cappelli et al., 2011) for indexing them. LSH emerges as a prime contender for high-dimensional spaces and addresses the nearest neighbor search (NNS) problem (Lu et al., 2018). Utilizing multiple hash functions, LSH maps a binary feature vector to natural numbers, which are then assigned to buckets and stored in the database.

3.2. Retrieval phase

During the retrieval phase, our objective is to identify palmprints closely resembling the latent query palmprints. Furthermore, it is imperative to maintain a compact candidate list.

Just as in the indexing phase, we acquired ROI, the OF, and minutiae for latent query palmprints. Following this, we generated feature vectors using the KHM-OKM method. To identify the most similar feature vectors stored in the database, we employed the equation labeled as Eq. (18). By assigning distinct weights to votes based on minutiae importance, we derived scores for the latent query palmprint in comparison to the palmprints stored in the database.

We employed the area-based registration method proposed by Krish et al. (2014) to refine the candidate list, utilizing it to reduce false

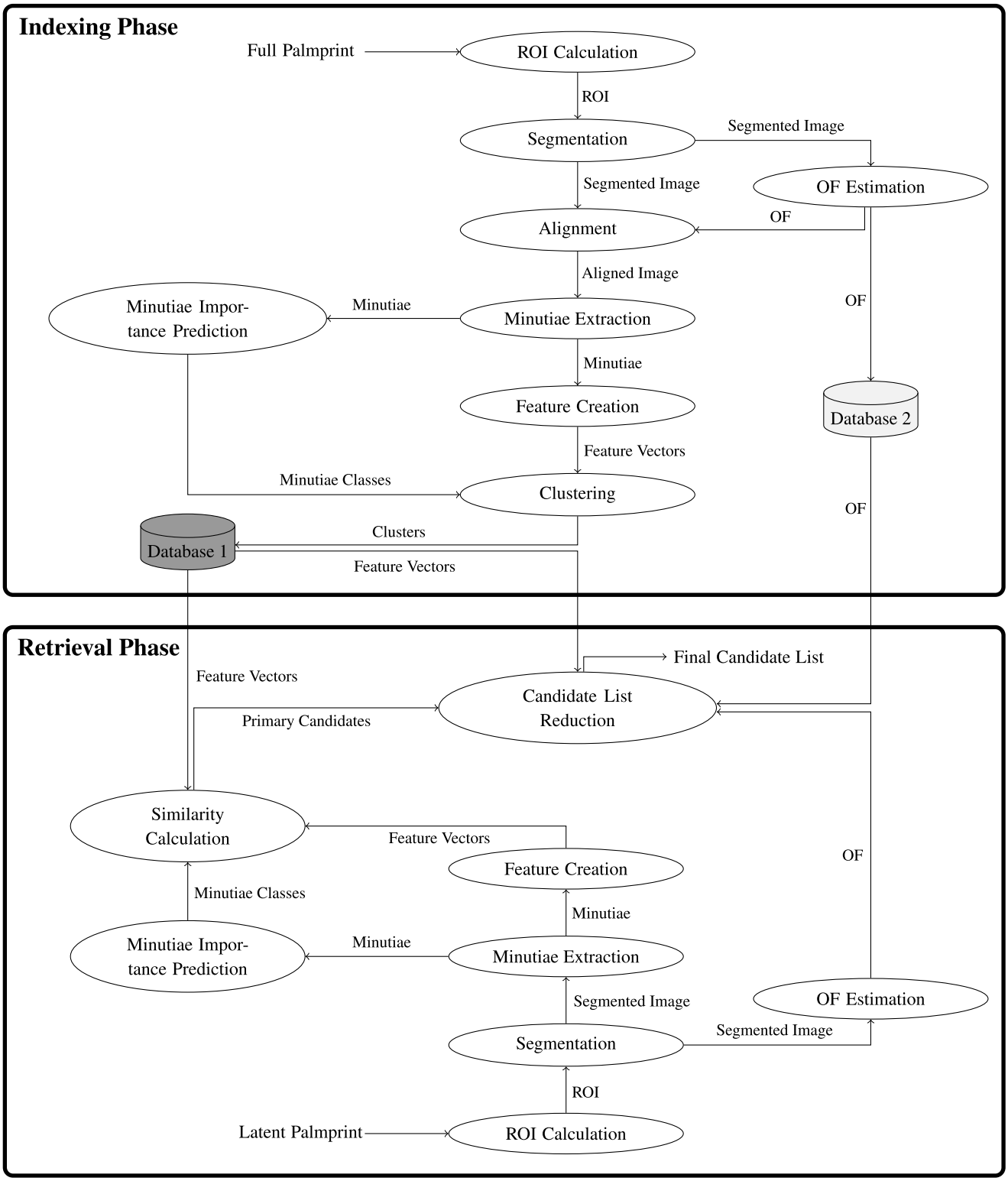


Fig. 9. Block diagram of the proposed indexing method. In this context, the acronyms OF and ROI stand for orientation field and region of interest, respectively.

positives and derive the final candidates. In our retrieval strategy, we compared the OF of the latent query palmprint with that of palmprints within the candidate list, identifying the most similar OFs. Aligning the latent palmprint with candidates having OF similarity exceeding a set threshold, we effectively filtered out candidates with insufficient similarity. After this filtering, feature vectors falling outside the region of the remaining palmprints were removed, leading to updated scores. For a visual representation, refer to Fig. 9 depicting the proposed method's block diagram.

Since we utilized the methodology introduced by Kavati et al. (2017) to make feature vectors, the procedure resembled the indexing phase. We systematically excluded certain extraneous minutiae from both the latent query palmprint and the palmprints within the sample image set, subsequently leading to the computation of matching scores. Following

Table 1
Matching scores between palmprints in the sample image set and the database, and with query palmprints. Bold highlights indicate the maximum value for each row in the database and the maximum value for each column in the query.

	Database			Query	
Sample Image Set	P000001_l	2_l_1	4_l_1	P000001_l_i_2	P000001_l_h_1
P000001_l_h	**107**	6	4	6	**10**
P000001_l_i	**33**	5	3	**23**	4
P000001_l_t	**7**	**7**	6	7	7

Table 2
Binarized matching scores between palmprints in the sample image set and the database, and with query palmprints.

	Database			Query	
Sample Image Set	P000001_l	2_l_1	4_l_1	P000001_l_i_2	P000001_l_h_1
P000001_l_h	1	0	0	0	1
P000001_l_i	1	0	0	1	0
P000001_l_t	1	1	0	0	0

this step, we normalized the matching scores and mapped them into the range of [0, 1] using the softmax function. Progressing from that point, we incorporated threshold considerations and applied LDA to convert the matching scores into binary form, thereby facilitating class separation. Subsequently, we identified candidates by applying hash functions to the feature vectors and tallying the occurrences of collisions with feature vectors placed in designated buckets. It is worth noting that certain methodologies, such as the one presented in (Ferrara et al., 2012), employed Eq. (20) to evaluate the similarity between two fixed-length binary feature vectors denoted as v_a and v_b.

$$s(v_a, v_b) = 1 - \frac{||v_a \; XOR \; v_b||_1}{n}, \tag{20}$$

where XOR represents a bitwise-exclusive-or operation, with n denoting the number of bits in each vector, and $||.||$ representing the 1-norm of a vector. However, we abstained from its usage due to the necessity of searching through all stored feature vectors in the database and computing the similarity between each pair of feature vectors. Furthermore, the binary feature vectors resulting from our improved version of Kavati et al.'s method exhibit a higher number of 0s compared to 1s. This discrepancy stems from the fact that for each latent query palmprint, the number of corresponding palmprints in the sample image set is fewer than those remaining in the set. Consequently, following the approach advocated in (Cappelli et al., 2011), we employed LSH for the indexing and retrieval of palmprints.

Allow us to delve into the specifics of the binarized iteration of Kavati et al.'s method (Kavati et al., 2017). Our database exclusively comprises full palmprints. Kavati et al.'s technique, however, necessitates a minimum of two full palmprints from the same palm. Regrettably, LPIDB v1.0 (Morales, Medina-Pérez et al., 2014), encompassing latent palmprints, partial impression palmprints, and full impression palmprints, lacks two full impression palmprints for a singular palm. As a result, we were prompted to adapt this methodology and formulate a strategy to accommodate this dataset. We incorporated all partial impression palmprints into the sample image set. This decision was primarily driven by the fact that each partial impression palmprint corresponds to a specific palm region, such as interdigital, thenar, or hypothenar. Much like partial impression palmprints, each latent palmprint is affiliated with one of the palm's regions.

To illustrate with a concrete example, consider the three full impression palmprints in the database: P000001_l in (Morales, Medina-Pérez et al., 2014), 2_l_1, and 4_l_1 in (THUPALMLAB). The sample image set comprises P000001_l_h, P000001_l_i, and P000001_l_t in (Morales, Medina-Pérez et al., 2014). Subsequently, we compute the matching score between each partial impression palmprint in the sample image set and the full palmprints stored in the database using the Bozorth3 matcher (NBIS). For our query palmprints, we select two latent palmprints: P000001_l_i_2 and P000001_l_h_1 from (Morales, Medina-Pérez et al., 2014), and derive the corresponding matching scores with the palmprints in the sample image set (see Table 1). Following this, we binarize the matching scores (see Table 2). Referring to Table 2, in the two right columns related to the query, we identify the matching scores that equaled 1. In the case of P000001_l_i_2, the associated palmprint in the sample image set is P000001_l_i. Consequently, within Table 2, we consider the second row of the database that is related to P000001_l_i and examine matching scores of 1. The only palmprint meeting this criterion is P000001_l. Similarly, for P000001_l_h_1, the linked palmprint in the sample image set is P000001_l_h. Within Table 2, the first row of the database that is related to P000001_l_h is considered, and all palmprints with matching scores of 1 are accounted for. Once again, the lone palmprint meeting this criterion is P000001_l. Consequently, this approach effectively retrieves P000001_l for both query palmprints P000001_l_i_2 and P000001_l_h_1.

Mirroring our indexing methodology, we employed the OF of the query palmprint and the OF of palmprints in the candidate list to locate a specific region within each full palmprint where its OF closely resembled that of the query palmprint. When this similarity surpassed a predetermined threshold, Krish et al.'s method (Krish et al., 2014) enabled the registration of the query palmprint into the full palmprint and facilitated the removal of minutiae situated outside this region. This process is illustrated in Fig. 10, which illustrates how genuine minutiae are retained on the corresponding full palmprint. Subsequently, erroneous minutiae and the triangles formed by them are eliminated, resulting in a reduction of the ER. As far as our knowledge extends, this method is the most suitable for registering partial prints. The pseudocodes for palmprint retrieval are detailed in Algorithms 2 and 3.

Lastly, the amalgamation of candidate lists was essential to derive the ultimate candidate list. To accomplish this, we referred to the approach outlined in Khodadoust et al.'s method (Khodadoust et al., 2022) and applied the technique introduced by Kabir et al. (2018) for candidate combination.

Our experiments validate the distinctiveness of our proposed feature vector, effectively retrieving a significant portion of the true candidates. In contrast to many voting methods that assign a weight of 1 to recognized feature vectors, we adopted a nuanced approach during the retrieval stage, assigning varying weights between 0 and 1 to votes. This strategic adjustment aims to curtail the ER. Furthermore, we improved the applicability of Kavati et al.'s method (Kavati et al., 2017), which initially could not be employed for latent and partial prints. By amalgamating it with the approach proposed by Krish et al. (2014), we effectively addressed this limitation and tailored the method to be compatible with latent and partial prints. Subsequent to this section, we will present the outcomes of our experimental endeavors. This report will encompass details regarding the utilized datasets, parameters, and software packages.

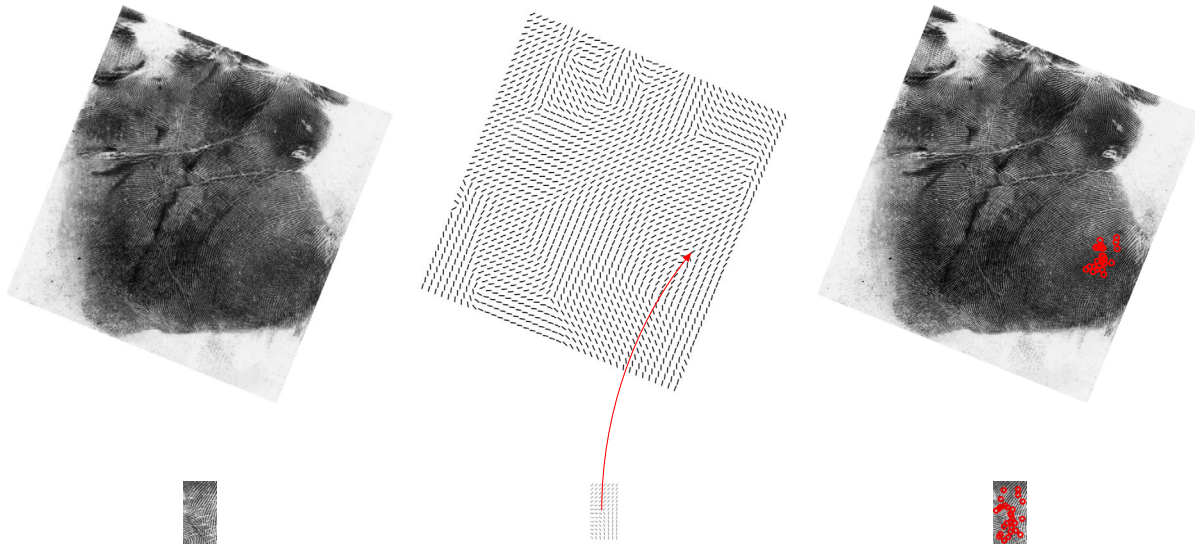


Fig. 10. The results of Krish et al.'s method (Krish et al., 2014) for selectively eliminating minutiae from corresponding full palmprints. The left images, from top to bottom, represent an aligned impression of a full palmprint (P000040_l in (Morales, Medina-Pérez et al., 2014)) and a latent palmprint (P000040_l_t_2 in (Morales, Medina-Pérez et al., 2014)). The middle images display the OF of the palmprint images in the left column. The right images depict minutiae extracted from the left images. The top right image reveals the remaining minutiae after the registration of the latent palmprint into the corresponding full palmprint using Krish et al.'s method.

4. Experiments

This section offers an in-depth overview of the palmprint databases employed in our experiments. Following that, we define the experimental setup, outlining the specific configurations and parameters utilized. Subsequently, we examine the accuracy measures employed in our experiments. Finally, we present a comprehensive analysis of the results, encompassing the evaluation of the proposed method's performance and its comparison with established approaches.

Algorithm 2 Pseudo-code for candidate retrieval.

Input: Latent query palmprint Q
Output: Candidate list L

$S \leftarrow \emptyset$;
$Segmented_Q \leftarrow Segmentation(Q)$;
$OF_Q \leftarrow Orientation\_Field(Segmented_Q)$; ▷ Using the Khan et al.'s method (Khan et al., 2016)
$M_Q \leftarrow Minutiae\_Extraction(Segmented_Q)$;
$V_Q \leftarrow Feature\_Vector(M_Q)$; ▷ Call Algorithm 1
$MI_Q \leftarrow Minutiae\_Importance(M_Q)$;
$V_{QW} \leftarrow Weight(V_Q(MI_Q))$; ▷ Assigning a weight to each feature vector V_Q according to minutiae importance
$C \leftarrow Searching(V_{QW}, D)$; ▷ Searching for each feature vector V_{QW} among the feature vectors stored in the database D during the indexing stage and generating the candidate list C
for each $c_i \in C$ **do**
  $M_R \leftarrow Similarity(OF_Q, OF_i)$; ▷ Using the Krish et al.'s method (Krish et al., 2014)
  if $M_R < t$ **then** ▷ t is a threshold
    $Remove(c_i)$;
  else
    $A_i \leftarrow Region(OF_Q, OF_i)$; ▷ The area within the full palmprint C where the OF closely resembles that of the latent query palmprint
    $V_R \leftarrow V_{QW}(A_i)$; ▷ If all three minutiae associated with a feature vector, which belongs to V_{QW}, are located within A_i, we will include the said feature vector in V_R
    $V_{Ri} \leftarrow Vote(V_R, V_i)$; ▷ V_i represents the feature vector with assigned weight corresponding to the palmprint within C
  $S_i \leftarrow Score(V_{Ri})$;
  $S \leftarrow S_i \cup S$;
$L \leftarrow Rank(S)$; ▷ Assigning ranks to candidates based on their scores
return L;

Algorithm 3 Pseudo-code for candidate retrieval using our improved version of Kavati et al.' method (Kavati et al., 2017).

Input: Latent query palmprint Q, sample image set SS
Output: Candidate list L_k

$V_{SS} \leftarrow \emptyset$;
$ii \leftarrow 1$;
$jj \leftarrow 1$;
$S \leftarrow \emptyset$;
$Segmented_Q \leftarrow Segmentation(Q)$;
$OF_Q \leftarrow Orientation\_Field(Segmented_Q)$; ▷ Using the Khan et al.'s method (Khan et al., 2016)
$M_Q \leftarrow Minutiae\_Extraction(Segmented_Q)$;
$MI_Q \leftarrow Minutiae\_Importance(M_Q)$;
for each $m_i \in M_Q$ **do**
  if $ii < m$ & $m_i \in RM_q$ **then** ▷ m represents the count of minutiae to be discarded, and RM_q signifies the removable minutiae within the query palmprint that exert a more pronounced impact on the matching score
    $Discard(m_i, MI_Q)$; ▷ Discarding the removable minutia
    $ii \leftarrow ii + 1$;
for each $Im_j \in SS$ **do** ▷ Im_j is a jth palmprint in SS
  $Segmented_j \leftarrow Segmentation(Im_j)$;
  $OF_j \leftarrow Orientation\_Field(Segmented_j)$;
  $M_j \leftarrow Minutiae\_Extraction(Segmented_j)$;
  $MI_j \leftarrow Minutiae\_Importance(M_j)$;
  for each $m_k \in M_j$ **do**
    if $jj < n$ & $m_k \in RM_j$ **then** ▷ n represents the count of minutiae to be discarded, and RM_j indicates the removable minutiae within SS that exert a more adverse effect on the matching score
      $Discard(m_k, MI_j)$; ▷ Discarding the removable minutiae
      $jj \leftarrow jj + 1$;
  $V_j \leftarrow Match(M_Q, M_j)$; ▷ Matching scores between Q and SS
  $V_{SS} \leftarrow V_{SS} \cup V_j$;
$C \leftarrow Searching(V_{SS}, D)$; ▷ Searching for each feature vector V_{SS} among the feature vectors stored in the database D during the indexing stage and generating the candidate list C
for each $Im_p \in C$ **do**
  $OF_p \leftarrow Orientation\_Field(Im_p)$;
  $S_p \leftarrow Similarity(OF_Q, OF_p)$; ▷ Assessing the similarity between OF_Q and the OF of each palmprint in the retrieved candidate list
  if $S_p < t$ **then** ▷ t is a threshold
    $Remove(Im_p)$;
  else
    $S \leftarrow S \cup S_p$;
$L_k \leftarrow Rank(S)$; ▷ Assigning ranks to candidates based on their scores
return L_k;

Fig. 11. Palmprint extraction: (a) synthesis handprint image (handprint_135_2 in (Morales, Cappelli et al., 2014)) and (b) manually extracted palmprint from the handprint image.

4.1. Datasets

- LPIDB v1.0 (Morales, Medina-Pérez et al., 2014): This dataset was created using 51 volunteers (28 males and 23 females) with ages ranging from 4 to 81. The dataset consists of a total of 788 palmprint images, including 380 latent palmprints, 102 full impression palmprints, and 306 partial impression palmprints. Consequently, each full impression palmprint typically encompasses more than one latent palmprint. The images possess varying sizes and are in BMP format.
- THUPALMLAB: This dataset was compiled using 80 volunteers, yielding a total of 1,280 palmprint images (eight impressions per volunteer's palm). All images are of the same size (2,040×2,040 pixels) and are formatted as BMP.
- Synthesis handprint images (Morales, Cappelli et al., 2014): The images in this dataset were synthetically generated in the form of handprints, featuring five fingerprints for fingertips, a palmprint, and a hand shape. A dataset of 5,000 synthetic handprint images was shared by Morales, Cappelli et al. (2014) for research purposes. Consequently, the total count of palmprint images is 5,000 (five impressions per hand). All images are of uniform size (3,200×3,200 pixels) and are formatted as JPG.

4.2. Experimental setup

Our approach involved the utilization of the MATLAB and Python programming languages. The experiments were conducted on a system equipped with an Intel® Core™ i7-7820HQ processor, 32 GB DDR4 RAM, and NVIDIA Quadro M1200 GPU.

In our experiments, we utilized three distinct datasets. Among them, LPIDB v1.0 (Morales, Medina-Pérez et al., 2014) comprises latent palmprints, employed as query palmprints in the retrieval phase This dataset encompasses full and partial impression palmprints. However, exclusively full impression palmprints were chosen for the indexing phase in our study. Furthermore, we incorporated data from THUPALMLAB and extracted palmprints from synthesis handprint images (Morales, Cappelli et al., 2014) to assess the resilience of our method against negative comparisons. It is worth noting that we manually extracted palmprints from the handprint images mentioned in (Morales, Cappelli et al., 2014). Fig. 11 provides an illustration of a palmprint extracted from a handprint image.

As the initial step of our approach involves palmprint segmentation, we initiated the process by transforming all palmprint images into the frequency domain, resulting in blurred binary images similar to Fig. 4(e). Subsequently, we partitioned the dataset into a training set and a testing set, reserving 80% of the data (6,715 images) for training and allocating 20% (353 images) for testing purposes. For the training phase, we applied augmentation techniques to enhance the available samples. For each image, we employed three levels of image resizing (resulting in 20,145 images), performed five random rotations (yielding 33,575 images), and executed horizontal flipping (resulting in 6,715 images). This collective effort culminated in a dataset comprising 67,150 palmprints. Following this, we resized all images to dimensions of 256×256. Within our methodology, we utilized the combination of two loss functions, BCE and Soft Dice losses, to compute the discriminator's loss during the training process. Note that the number of epochs in our model was set to 30 to achieve convergence.

According to Eq. (5), it is necessary to acquire the optimal values for W_b and W_s. Additionally, determining the optimal value for the Delaunay triangulation of order k and the preliminary consideration of the optimal value for the neighborhood size parameter λ is also crucial.

In this study, we applied machine learning techniques to identify the optimal values for all parameters, treating these variables as input parameters to generate the desired output. The random search (RS) method, introduced by Bergstra and Bengio (2012), was employed to ascertain the optimal values for these variables.

In our experimental trials, the optimal values for W_b and W_s were determined as 0.57 and 0.43, respectively. Moreover, the optimal value for the order k in Delaunay triangulation was found to be 10. Additionally, the optimal value for the neighborhood size parameter λ was established as 6. As λ increased from 0 to 8, the ER diminished, while PR rose. The optimal value for λ is the juncture at which $ER \approx PR$.

Another crucial parameter for our method is p in Eqs. (16) and (17). The optimal value for this parameter was determined using the RS method and is set to 3. For Eq. (13), we set α to 0 and β to 1 for normalizing feature values.

Our minutiae importance method, as well as our improved version of Kavati et al.'s method (Kavati et al., 2017), requires minutiae matching. For this purpose, we utilized two minutiae matchers: MCC (Cappelli et al., 2010) and Bozorth3 (NBIS).

In our improved version of Kavati et al.'s method (Kavati et al., 2017), a strategic refinement involved the exclusion of up to 15 removable minutiae from both query palmprints and palmprints within the sample image set. For full palmprints, we discarded a maximum of 45 removable minutiae. These values represent the optimal number of removable minutiae. Complete elimination of all removable minutiae could adversely affect matching scores; thus, identifying the optimal quantity for removal is imperative.

Considering that each feature vector in our improved version of Kavati et al.'s method (Kavati et al., 2017) comprises 306 dimensions, we utilized 32 hash functions, each selecting 24 bits. We excluded query feature vectors with only 0 bits and eliminated candidates with entirely 0-bit feature vectors. Additionally, we applied the pruning method recommended by Krish et al. (2014) to further refine candidates.

Note that for our experiments, we used two palmprint sets: a training set and a test set. The training set comprises 80% of the palmprints, while the test set contains the remaining 20%. The training set was employed during the training stage, and upon completion of training on this set, we evaluated the accuracy of our model using the test set.

4.3. Accuracy measures

The accuracy of palmprint indexing approaches is typically assessed by considering the trade-off between the ER and the penetration rate (PR). The ER signifies the percentage of searched palmprints that remain unfound, while the PR indicates the average portion of the database that the system needs to search.

These metrics play a critical role in evaluating the effectiveness of a palmprint indexing system. A well-designed system should strive for a low ER, minimizing both incorrect matches and missed matches, while concurrently aiming for a high PR, indicating efficient coverage of a substantial portion of the database. A high PR implies that, on average, the system efficiently searches only a small fraction of the database to

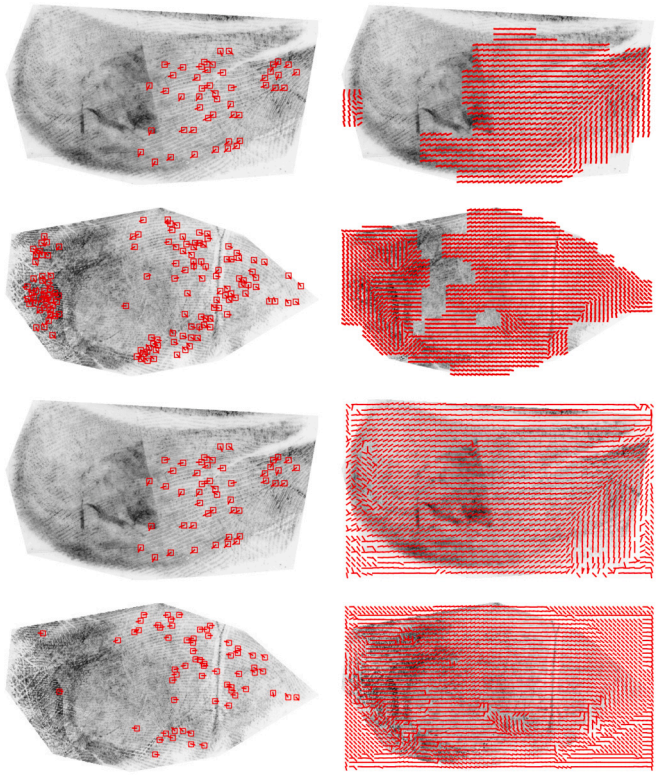


Fig. 12. Comparison of minutiae and the OF obtained for two latent palmprints using two distinct methods: In the top left and immediately below it, respectively, display the minutiae extracted from P000008_l_h_1 and P000008_r_t_1, as presented in (Morales, Medina-Pérez et al., 2014), employing the MinutiaeNet (Nguyen et al., 2018). On the right of these images, the corresponding OFs are illustrated, estimated using the MinutiaeNet. The images in the two bottom rows showcase both minutiae and OFs through the utilization of FingerNet (Tang et al., 2017).

identify an individual, facilitating the quick narrowing down of potential matches.

4.4. Results

Our initial experiment focused on evaluating our segmentation method and comparing it with existing palmprint segmentation methods. It is essential to note that some deep convolutional neural network (DCNN)-based methods, such as FingerNet (Tang et al., 2017) and MinutiaeNet (Nguyen et al., 2018), have demonstrated accurate segmentation for fingerprints. These methods excel at estimating the OF and extracting genuine minutiae from fingerprints. However, it is crucial to emphasize that these DCNN-based methods designed for fingerprints do not perform optimally for palmprints.

In Fig. 12, we showcase minutiae and the OF obtained for two latent palmprints using FingerNet and MinutiaeNet. Despite FingerNet yielding better results than MinutiaeNet, we refrain from directly comparing our segmentation method with these fingerprint-based techniques. The reason being that FingerNet and MinutiaeNet are trained specifically for fingerprints, and their accuracy diminishes when applied to palmprints. It would be unfair to assess them against our method, tailored for palmprints. Consequently, our comparative analysis is solely focused on palmprint-based methods.

To compare our segmentation method with other methods, we focused on the genuine minutiae extracted after segmentation. It is evident that the accuracy improves with a higher count of genuine minutiae. Fig. 13 illustrates the percentage of genuine minutiae, while Fig. 14 compares the accuracy achieved by the extracted minutiae using these methods in our indexing approach.

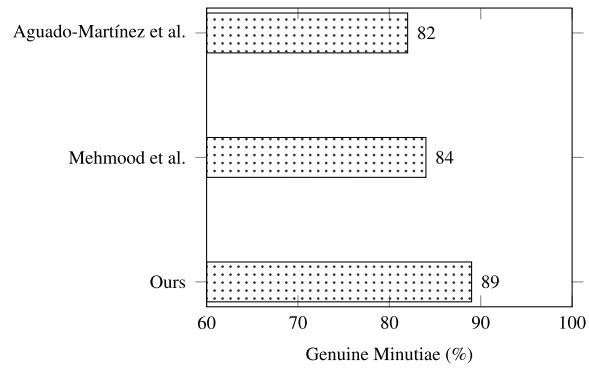


Fig. 13. Comparison of correctly extracted minutiae after segmentation by our method, Mehmood et al. (2023), and Aguado-Martínez et al. (2021).

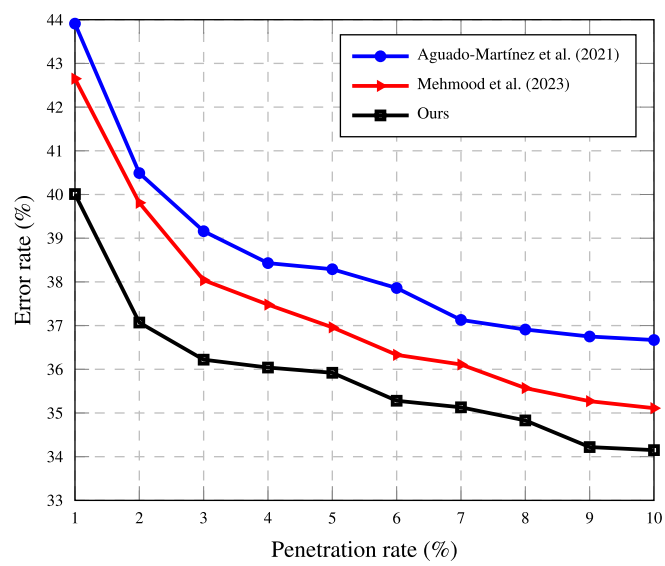


Fig. 14. Comparing our indexing method with Mehmood et al. (2023), and Aguado-Martínez et al. (2021). Demonstrating the superior accuracy of our method.

The next experiment was conducted to assess the impact of the OF on our indexing method. We compared two methods, namely Khan et al. (2016) and Kovesi (2022). Fig. 15 demonstrates that Khan et al.'s method outperforms Kovesi's method, making it more suitable for integration into our approach.

Following this, the next two experiments were conducted to predict minutiae importance. As mentioned earlier, we modified the method proposed by Loyola-González et al. (2021) for predicting minutiae importance, incorporating a feature known as MiCI. With the aim of classifying minutiae into six distinct classes, we adapted binary classifiers to accommodate the multi-class nature of the problem. Specifically, we utilized three classifiers: multi-class linear discriminant analysis (MCLDA), multi-class random forest (MCRF), and multi-class logistic regression (MCLR). The outcomes produced by these classifiers are compared in Fig. 16. Consequently, we selected the MCLDA classifier for predicting minutiae importance.

In our subsequent experiment, we compared the minutiae importance derived from the methods of Makni and Charrier (2020), Loyola-González et al. (2021), and our modified version of Loyola-González et al.'s method. We explored the impact of these approaches on our indexing method, as depicted in Fig. 17.

In our following experiment, we aimed to select an appropriate minutiae matcher. We evaluated MCC (Cappelli et al., 2010) and Bozorth3 (NBIS) matchers and observed that the use of MCC matcher

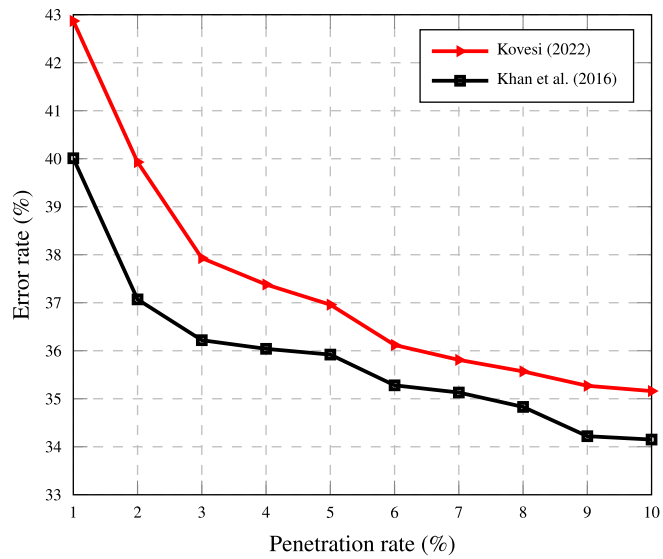


Fig. 15. The impact of the OF proposed by Khan et al. (2016) and Kovesi (2022) on our indexing method.

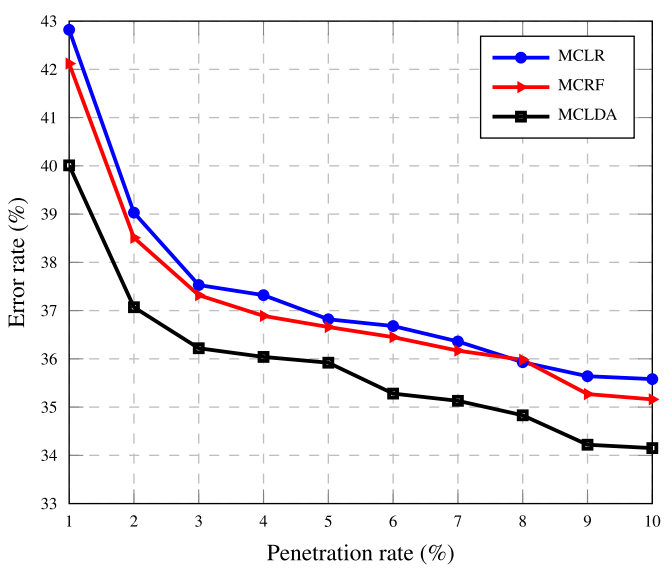


Fig. 16. The results of our indexing method using three distinct classifiers within our modified version of Loyola-González et al.'s approach.

resulted in a time-consuming matching process (see Table 3). The MCC method involves the consideration of a cylinder encompassing each minutia and its neighboring minutiae, with the similarity between cylinders indicating the degree of match. Consequently, the removal of a minutia impacts all cylinders involving that minutia. Given that numerous minutiae were categorized as removable when employing MCC matcher, we opted to utilize solely the Bozorth3 matcher in our indexing method, as well as in our improved version of Kavati et al.'s method (Kavati et al., 2017).

Furthermore, we conducted full palmprint alignment prior to minutiae extraction, a prerequisite for the application of Krish et al.'s method (Krish et al., 2014). Notably, we observed that the rotation of full palmprints had an impact on the importance of minutiae. This implies the necessity to align full palmprints and harmonize them within a common coordinate system. Fig. 18 illustrates the minutiae classes generated by our method using both MCC and Bozorth3 matchers, highlighting the impact of full palmprint alignment on minutiae importance.

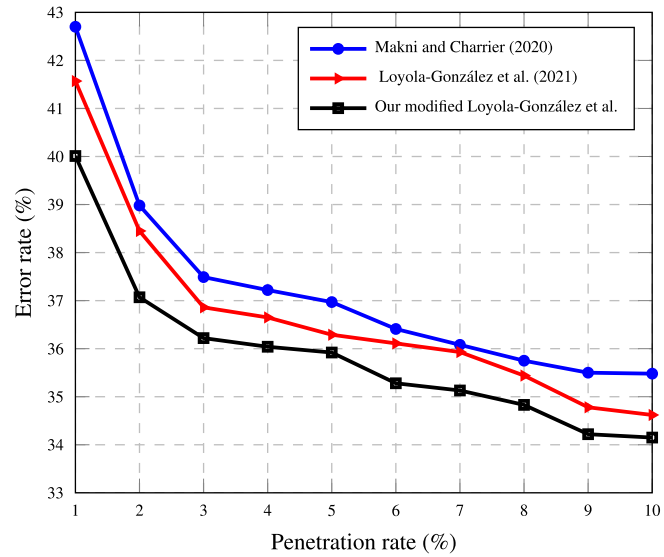


Fig. 17. The outcomes of our indexing method employing the approach by Makni and Charrier (2020), the technique proposed by Loyola-González et al. (2021), and our modified version of the Loyola-González et al.'a method.

Table 3
Average time in second for Bozorth3 (NBIS) and MCC (Cappelli et al., 2010).

Bozorth3	MCC
0.27	75.41

The figure demonstrates that MCC matcher is more robust to rotation compared to the Bozorth3 matcher.

Fig. 19 presents the outcomes of minutiae importance determination in our improved version of Kavati et al.'s method (Kavati et al., 2017) for both MCC and Bozorth3 matchers. In this figure, 'removable ridge bifurcations' and 'removable ridge endings' indicate the percentage of minutiae that, when discarded, lead to an increase in matching scores. Conversely, 'irremovable ridge bifurcations' and 'irremovable ridge endings' correspond to the percentage of minutiae that, if removed, result in a decrease in matching scores. As such, they must be retained. 'Neutral ridge bifurcations' and 'neutral ridge endings' denote the percentage of minutiae whose removal has no impact on matching scores.

As observed, the proportion of removable ridge bifurcations is approximately 27.134% for MCC and 1.085% for Bozorth3. Similarly, the percentages for removable ridge endings are approximately 25.989% for MCC and 0.976% for Bozorth3. Additionally, the percentages of irremovable ridge bifurcations are around 22.139% for MCC and 1.080% for Bozorth3. Likewise, the percentages for irremovable ridge endings are approximately 23.15% for MCC and 1.464% for Bozorth3. These results highlight a discernible relationship between minutiae importance and their respective types. Notably, the proportion of removable ridge bifurcations surpasses that of removable ridge endings for both matchers. Conversely, the proportion of irremovable ridge bifurcations is lower than that of irremovable ridge endings in both cases. The greater number of removable ridge bifurcations compared to removable ridge endings, and the lower number of irremovable ridge bifurcations compared to irremovable ridge endings, indicate that a majority of ridge bifurcations are removable. Removing them has the potential to enhance the matching score. Consequently, ridge bifurcations are better candidates for removal from the minutiae list.

In our subsequent experiment, we eliminated weights, ranging between 0 and 1, assigned to feature vectors in the voting mechanism

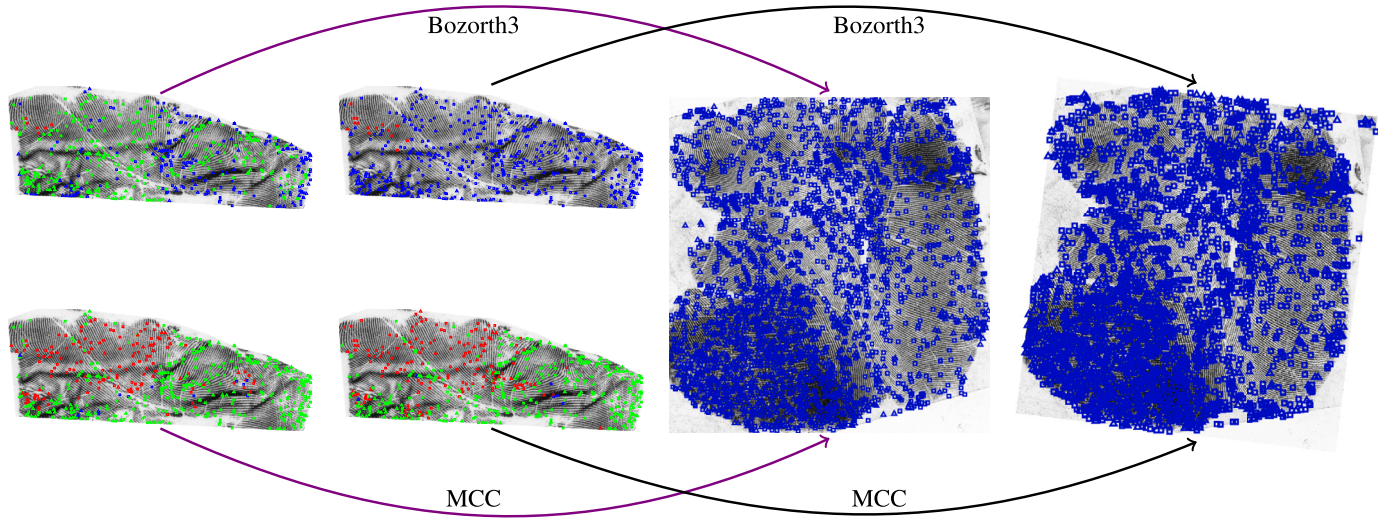


Fig. 18. Assessing of minutiae importance for a latent palmprint (P000007_r_i_1 in (Morales, Medina-Pérez et al., 2014)) using MCC (Cappelli et al., 2010) and Bozorth3 (NBIS) matchers. The illustration includes two full palmprints on the right: P000007_r from (Morales, Medina-Pérez et al., 2014), captured before and after alignment. In this representation, green, red, and blue correspondingly indicate removable, irremovable, and neutral minutiae (full color figure available online). Additionally, circular and square markers represent ridge endings and ridge bifurcations. Notably, the analysis demonstrates that MCC identifies a greater count of removable and irremovable minutiae compared to Bozorth3. Moreover, aligning the corresponding full palmprint image significantly impacts the performance of Bozorth3.

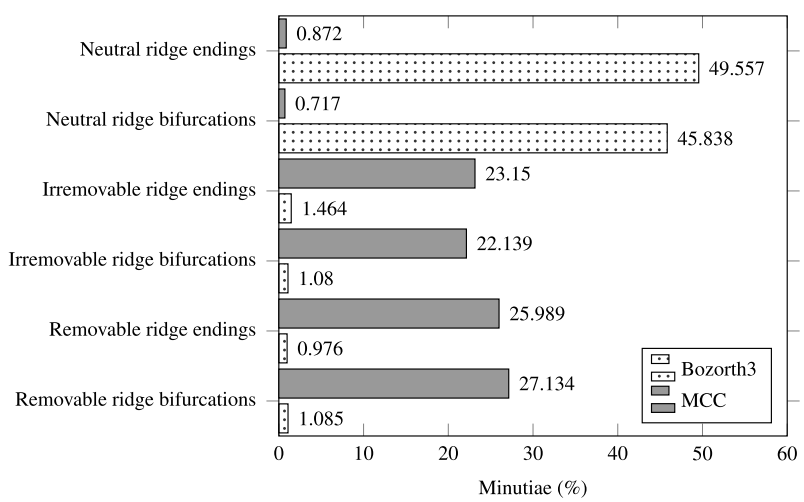


Fig. 19. Assessing minutiae importance using two different matchers. 'Removable' ridge bifurcations and endings refer to minutiae whose exclusion results in an enhanced matching score. Conversely, 'irremovable' ridge bifurcations and endings pertain to minutiae that, if omitted, lead to a reduced matching score, thus necessitating their retention. Lastly, 'neutral' ridge bifurcations and endings denote minutiae whose inclusion or exclusion has no impact on the matching score.

used for calculating the final score. In our indexing method, different values of w were considered for each ellipse within the query palmprint and its corresponding counterpart in the database. Fig. 20 illustrates the impact of removing weights from feature vectors based on minutiae importance in the context of the ER.

In Fig. 21, we demonstrate the impact of omitting removable and irremovable minutiae on the ER for our improved version of Kavati et al.'s method (Kavati et al., 2017) along with Krish et al. (2014). The illustration highlights the average ER reduction of 0.732% through the removal of removable minutiae.

In the next experiment, we explored the impact of choosing ellipses instead of triangles. We do not assert that ellipses are universally better than triangles for palmprint indexing; rather, their effectiveness depends on the features we choose for our feature vectors. Two palmprint indexing methods are based on triangles, and we compared our method with them. Fig. 22 illustrates the comparison, demonstrating that our method outperforms them.

In the last experiment, we combined the candidates identified through our feature vectors with those obtained using our improved version of Kavati et al.'s method (Kavati et al., 2017). Subsequently, we benchmarked this composite approach against three of the most proficient methods for palmprint indexing (see Fig. 23). In essence, a superior palmprint indexing method will exhibit a lower ER while maintaining the same PR as other methods. The data presented in this figure makes it evident that our method outperforms other approaches, demonstrating consistently lower ER across various PR values.

Tables 4 and 5 present the average execution time of the proposed method.

5. Conclusion

In this paper, we introduced a novel segmentation approach applicable to both latent and impression palmprints. Leveraging frequency domain analysis and a GAN, our method efficiently delineates ROI. Fur-

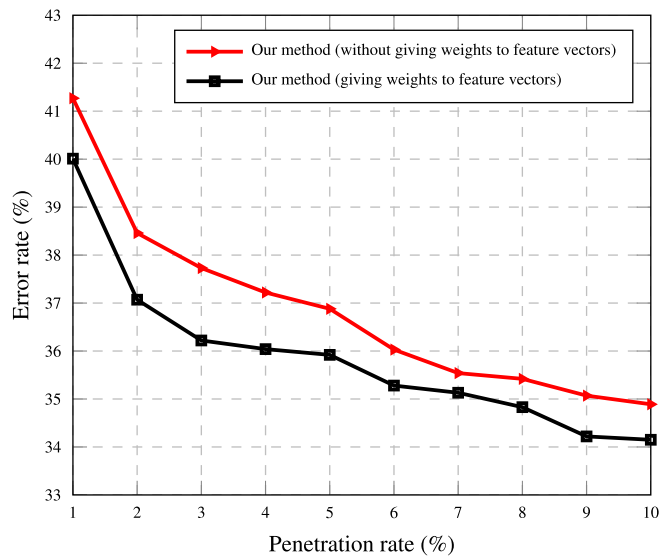


Fig. 20. The impact of removing weights from feature vectors based on minutiae importance on the ER, illustrating an increase in the ER by approximately 0.964% on average.

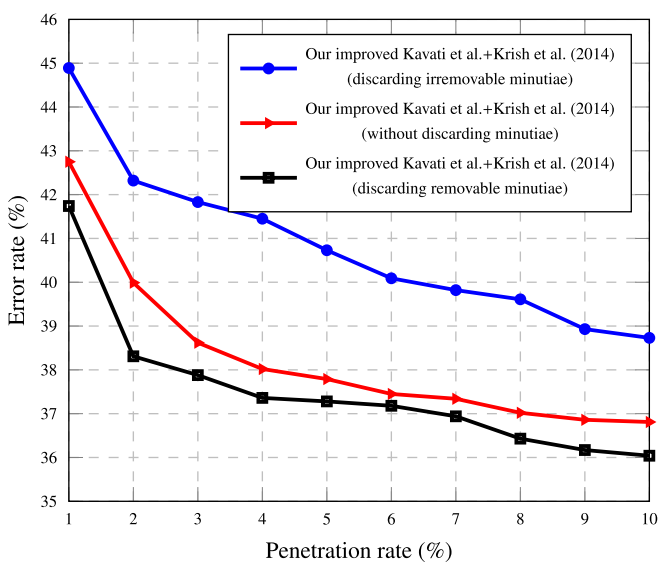


Fig. 21. The impact of omitting removable and irremovable minutiae on ER. Illustrating ER reduction through removal of removable minutiae (approximately 0.732% on average).

Table 4
Average execution time (in seconds) of the preprocessing methods used in our method to extract minutiae.

Operation	Latent palmprints	Full palmprints
Segmentation	0.298	1.179
OF estimation	0.057	0.558
Orientation diffusion	80.682	246.762
Enhancement	0.442	2.689

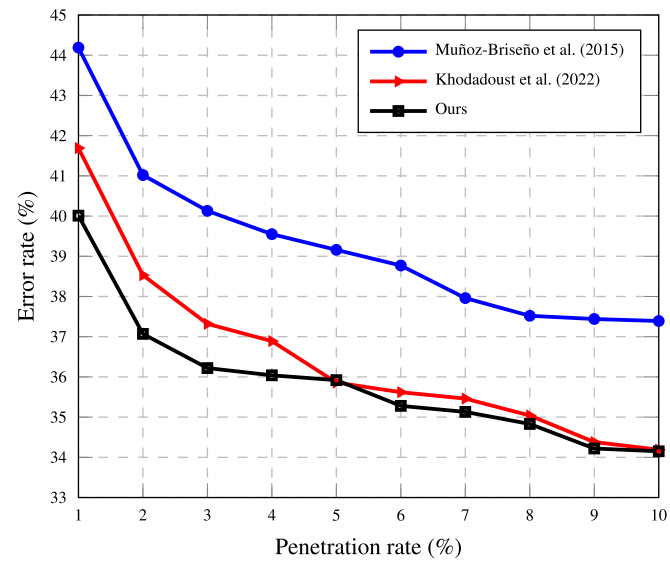


Fig. 22. Comparison of our method with other triangle-based methods, highlighting the superior accuracy of our approach.

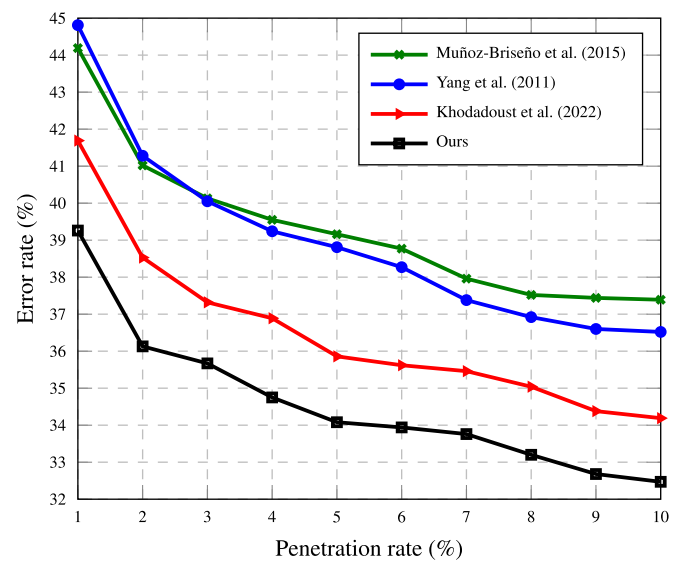


Fig. 23. Comparison of our method (combining our indexing method with the improved version of Kavati et al. (2017)) with other approaches, highlighting its superior accuracy.

Table 5
Average execution time (in seconds) for various steps in palmprint retrieval.

Operation	Time
Feature extraction	2.298
Computation of the scores and normalization	0.128
Fusion at score level	0.223
Candidate list reduction	0.119
Search	0.164

thermore, we presented a pioneering latent palmprint indexing scheme centered on minutiae. This marks the first endeavor to incorporate minutiae importance in palmprint indexing. Our proposed feature vector, derived from ellipse properties, capitalizes on Delaunay triangulation to produce three ellipses for each triangle. During retrieval, we implemented the registration method devised by Krish et al. (2014) to eliminate false-positive feature vectors akin to our latent query palmprints. Additionally, by assigning varying weights to retained feature vectors to compute scores, we effectively mitigated the number of false candidates. This strategic intervention resulted in a notable 0.964% reduction in the ER.

Moreover, we improved the score-based palmprint indexing framework presented by Kavati et al. (2017). Integrating the registration method from Krish et al. (2014) enabled more precise matching scores

and the removal of false candidates. To further optimize performance, certain minutiae that negatively impacted matching scores were prudently excluded. The cumulative effect of these refinements contributed to an ER reduction of approximately 0.732%.

Moving forward, our future endeavors entail developing a refined technique for enhancing latent palmprints. The limited availability of suitable latent palmprint samples has motivated our pursuit of larger datasets or synthetic generation, which we envisage as an ongoing avenue of research. Furthermore, our agenda includes the development of a faster and more accurate method for predicting minutiae importance, addressing an essential aspect of our current approach.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors extend their deep appreciation to the editors and the four anonymous reviewers for their invaluable comments and suggestions, significantly enhancing the content and presentation of this paper. Special gratitude is extended to Dr. Aythami Morales from the Universidad Autónoma de Madrid for generously providing synthesis handprint images. Furthermore, this research received partial funding from the National Council of Science and Technology of Mexico (CONACYT) under the scholarship grant 1154958.

References

Aguado-Martínez, M., Hernandez-Palancar, J., Castillo-Rosado, K., Cupull-Gómez, R., Kauba, C., Kirchgasser, S., & Uhl, A. (2021). Document scanners for minutiae-based palmprint recognition: a feasibility study. *Pattern Analysis & Applications*, *24*, 459–472.

Arora, G., Kalra, S., Bhatia, A., & Tiwari, K. (2021). PalmHashNet: palmprint hashing network for indexing large databases to boost identification. *IEEE Access*, *9*, 145912–145928.

Balwant, M. K., Agarwal, A., & Rao, C. R. (2015). Online touchless palmprint registration system in a dynamic environment. *Procedia Computer Science*, *54*, 799–808.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(1), 281–305.

Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., & Blaschko, M. B. (2019). Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *International conference on medical image computing and computerassisted intervention*. Springer.

Cappelli, R., Ferrara, M., & Maltoni, D. (2010). Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(12), 2128–2141.

Cappelli, R., Ferrara, M., & Maltoni, D. (2011). Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 1051–1057.

Ferrara, M., Maltoni, D., & Cappelli, R. (2012). Noninvertible minutia cylinder-code representation. *IEEE Transactions on Information Forensics and Security*, *7*(6), 1727–1737.

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE*, *11*(4), 1–31.

Jain, A. K., & Feng, J. (2009). Latent palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(6), 1032–1047.

Kabir, W., Ahmad, M. O., & Swamy, M. N. S. (2018). Normalization and weighting techniques based on genuine-impostor score fusion in multi-biometric systems. *IEEE Transactions on Information Forensics and Security*, *13*(8), 1989–2000.

Kavati, I., Prasad, M. V. N. K., & Bhagvati, C. (2017). Efficient score-based indexing technique for fast palmprint retrieval. In *Efficient biometric indexing and retrieval techniques for large-scale systems*. Springer.

Khan, M. A. U., Khan, T. M., Bailey, D. G., & Kong, Y. (2016). A spatial domain scar removal strategy for fingerprint image enhancement. *Pattern Recognition*, *60*, 258–274.

Khanmohammadi, S., Adibeig, N., & Shanehbandy, S. (2017). An improved overlapping k-means clustering method for medical applications. *Expert Systems with Applications*, *67*, 12–18.

Khodadoust, J., & Khodadoust, A. M. (2017a). Fingerprint indexing based on minutiae pairs and convex core point. *Pattern Recognition*, *67*, 110–126.

Khodadoust, J., & Khodadoust, A. M. (2017b). Fingerprint indexing based on expanded Delaunay triangulation. *Expert Systems with Applications*, *81*, 251–267.

Khodadoust, J., Medina-Pérez, M. A., Loyola-González, O., Monroy, R., & Khodadoust, A. M. (2022). A secure and robust indexing algorithm for distorted fingerprints and latent palmprints. *Expert Systems with Applications*, *206*, Article 117806.

Kovesi, P. D. (2022). MATLAB and Octave functions for computer vision and image processing. Retrieved from. https://www.peterkovesi.com/matlabfns.

Krish, R. P., Fierrez, J., Ramos, D., Ortega-Garcia, J., & Bigun, J. (2014). Partial fingerprint registration for forensics using minutiae-generated orientation field. In *2nd IEEE international workshop on biometrics and forensics, Malta*.

Li, W., You, J., & Zhang, D. (2005). Texture-based palmprint retrieval using a layered search scheme for personal identification. *IEEE Transactions on Multimedia*, *7*(5), 891–898.

Liu, E., Jain, A. K., & Tian, J. (2013). A coarse to fine minutiae-based latent palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(10), 2307–2322.

Liu, M., & Qian, P. (2021). Automatic segmentation and enhancement of latent fingerprints using deep nested unets. *IEEE Transactions on Information Forensics and Security*, *16*, 1709–1719.

Loyola-González, O., Ferreira Mehnert, E. F., Morales, A., Fierrez, J., Medina-Pérez, M. A., & Monroy, R. (2021). Impact of minutiae errors in latent fingerprint identification: assessment and prediction. *Applied Sciences*, *11*(9), 4187.

Lu, K., Wang, H., Xiao, Y., & Song, H. (2018). Why locality sensitive hashing works: a practical perspective. *Information Processing Letters*, *136*, 49–58.

Makni, L., & Charrier, C. (2020). Minutia confidence index: a new framework to qualify minutia usefulness. In *IEEE international conference on cyberworlds (CW)*.

Maltoni, D., Maio, D., Jain, A. K., & Feng, J. (2021). *Handbook of fingerprint recognition* (3rd ed.). Springer.

Mehmood, A. B., Taj, I. A., & Ghafoor, M. (2023). Palmprint enhancement network (PEN) for robust identification. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-16043-z.

Morales, A., Cappelli, R., Ferrer, M. A., & Maltoni, D. (2014). Synthesis and evaluation of high resolution hand-prints. *IEEE Transactions on Information Forensics and Security*, *9*(11), 1922–1932.

Morales, A., Medina-Pérez, M. A., Ferrer, M. A., García-Borroto, M., & Robles, L. A. (2014). LPIDB v1.0 - Latent palmprint identification database. In *IEEE international joint conference on biometrics (IJCB)*.

Muñoz-Briseño, A., Hernández-Palancar, J., & Gago-Alonso, A. (2015). *Minutiae based palmprint indexing*. *In Lecture notes in computer science: Vol. 8915*. Japan Springer.

NBIS, https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis. (Accessed 16 September 2022).

Nguyen, D. L., Cao, K., & Jain, A. K. (2018). Robust minutiae extractor: integrating deep networks and fingerprint domain knowledge. In *IEEE international joint conference on biometrics (IJCB)*.

Paliwal, A., Jayaraman, U., & Gupta, P. (2010). A score based indexing scheme for palmprint databases. In *IEEE international conference on image processing*.

Peralta, D., Galar, M., Triguero, I., Paternain, D., García, S., Barrenechea, E., Benítez, J. M., Bustince, H., & Herrera, F. (2015). A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation. *Information Sciences*, *315*, 67–87.

Schuch, P. (2019). Survey on features for fingerprint indexing. *IET Biometrics*, *8*(1), 1–13.

Soleimani, H., & Ahmadi, M. (2018). Fast and efficient minutia-based palmprint matching. *IET Biometrics*, *7*(6), 573–580.

Statista, https://www.statista.com/statistics/234785/international-tourists-arrivals-in-china/. (Accessed 16 September 2022).

Sutthiwichaiporn, P., & Areekul, V. (2013). Adaptive boosted spectral filtering for progressive fingerprint enhancement. *Pattern Recognition*, *46*(9), 2465–2486.

Tang, Y., Gao, F., Feng, J., & Liu, Y. (2017). FingerNet: an unified deep network for fingerprint minutiae extraction. In *IEEE international joint conference on biometrics (IJCB)*.

THUPALMLAB, http://ivg.au.tsinghua.edu.cn/dataset/THUPALMLAB.php. (Accessed 16 September 2022).

VeriFinger, https://www.neurotechnology.com/verifinger.html. (Accessed 16 September 2022).

Villarino, M. B. (2005). Ramanujan's perimeter of an ellipse. Retrieved from. arXiv:math/0506384.

Wang, R., Ramos, D., & Fierrez, J. (2011). Latent-to-full palmprint comparison based on radial triangulation under forensic conditions. In *IEEE international joint conference on biometrics (IJCB)*.

Worldometer, https://www.worldometers.info/world-population/china-population/. (Accessed 16 December 2022).

Yang, X., Feng, J., & Zhou, J. (2011). Palm print indexing based on ridge features. In *IEEE international joint conference on biometrics (IJCB)*.

Zhao, F., & Tang, X. (2007). Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. *Pattern Recognition*, *40*(4), 1270–1281.

Zhou, Y., Guo, T., Wu, M., & Zhao, T. (2010). Latent palmprint image segmentation based on dissimilarity tolerance. In *IEEE international conference on multimedia communications*.